Financial Cost Sensitive Feature Selection Method Based on Association Rule Applied to Credit Scoring

Ghislain Dorian Tchuente Mondjo, Kely Maxime Motue Djoko

Financial Cost sensitive Feature Selection Method based on Association rule applied to credit scoring

Ghislain Dorian Tchuente Mondjo[1,3]
tchuente.mondjo@gmail.com and
Kely Maxime Motue Djoko[1,2,3]
motuekely@gmail.com

[1] Université de Yaoundé I, Faculté des Sciences, Département d'Informatique
BP 812 Yaoundé, Cameroun
[2] IRD, Sorbonne Université, UMMISCO
F-93143, Bondy, France
[3] Fondation pour la Recherche l'Ingénierie et l'Innovation (FR2I)
BP 14306 Yaoundé, Cameroun

Abstract. Cost sensitive machine learning techniques are frequently used by data mining researchers, to improve their model. In banking credit scoring in particular, the financial cost of misclassification leads us to pay particular attention to cost-sensitive modelling. Knowing that the feature selection is a crucial preprocessing step in machine learning because it greatly affects a model's capacity for prediction and generalization, it becomes important to propose financial cost sensitive feature selection methods. In this paper, we propose Cost Sensitive Association Rules Feature Selection more Larger (ARFSL-CS), which is a two-phase feature selection algorithm based on Sequential Forward Selection (SFS). In the first phase, SFS selects the attributes according to a confidence threshold of the association rules sensitive to the financial cost whose consequent is the class of the loan and in which they are in the antecedent. In the second phase, SFS selects from the remaining features those which minimize the financial cost of misclassification of the model when applied to a dataset described only by each of the features. Experiments carried out on three datasets, using a cost sensitive Gradient Boosting credit scoring model show that ARFSL-CS selects the attribute subsets that most minimize the overall cost of financial misclassification for the banker with a gain of 8.7%, 21% and 69.9% compared to the prediction model without feature selection in the three considered datasets.

Keywords: Cost-sensitive feature selection · Association rules · credit scoring · financial cost

1 Introduction

From 2023, the Basel III approach [4] encourages to revise credit risk framework in banking lending process. In fact, the recent financial crisis and regulation concern made credit scoring methods a major topic in the banking industry. Credit risk analysis will use datamining methods to extract useful knowledge from bank's data to identify potential bad loans. One piece of knowledge that could be useful to the bank is the set of lender attributes that helps better distinguish between paid and unpaid loans. To this end, the selection of the most relevant attributes has become a key theme in the process of building prediction models.

Because it greatly affects a model's capacity for prediction and generalization, attribute selection is a crucial preprocessing step in machine learning. A training set with a high dimensionality will provide a model that is hard to interpret, so attribute selection can be used to either create a model with greater prediction performance or a more interpretable model.

Large number of works have been done in the literature [18] on feature selection process. These methods can be classified into three main categories: Filter, Wrapper, and Embedded. A common wrapper-type method used is Sequential feature selection (SFS). SFS [5] is a greedy algorithm that iteratively adds or removes features from a dataset in order to improve the performance of a predictive model. SFS can be either forward selection or backward selection. The LASSO [3] and Ridge [4] regularization approaches are common embedded methods that tackle the limit of comon approaches that is greater computational cost.

On the other hand, cost-sensitive learning is a subfield of machine learning that addresses classification problems where the misclassification costs are not equal [25]. It is common for data mining researchers to use cost-sensitive machine learning methods that incorporate the costs of classification error in their construction [26]. Cost-sensitive learning helps solve problems often related to the class imbalance problem and an important cost related to business criteria is the financial cost used by [22] which shows that cost-sensitive algorithms can better increase the retailer profits compared to cost-insensitive algorithms.

The purpose of this work is to know if it is possible to increase bank's profit by using a cost sensitive algorithm build from the most relevant attributes choosen also in a cost sensitive way. Cost-sensitive automatic classification techniques rely on data that usually con-

[4] https://www.bis.org/bcbs/publ/d570_highlights.htm

tains redundancies or even irrelevant attributes to minimize financial losses. Removing redundancies or eliminating attributes that are irrelevant from a financial cost sensitivity perspective can be helpful in reducing the banker's losses and shortfalls.

In order to take in consideration the financial costs during the selection of attributes, this paper proposes a Cost Sensitive Association Rules Feature Selection more Large (ARFSL-CS) method that select attributes that minimize the banker's financial losses. The idea here is to integrated financial misclassification cost to compute the score attribute vector that classify the attributes according to their relevance. The objective of ARFSL-CS is to select the most significant subset of attributes for the bank and use it to construct a cost-sensitive algorithm that will reduce the banker's losses.

The remainder of this paper is arranged as follows : In Section 2, we present related work. The proposed methodology is presented in Section 3. The experimental results of the proposed model are presented in section 4. Conclusion and suggestions for further research are discussed in Section 5.

2 Related Work

The related work of this paper will be divided in four parts: Feature selection, Association rule Feature selection method, cost sensitive feature selection methods and Financial cost sensitive method in datamining.

2.1 Feature selection

Feature selection is the process of isolating the most consistent, non-redundant, and relevant features to use in model construction. As the quantity and diversity of datasets increase, it is crucial to gradually reduce their size. Reducing modeling's computational cost and enhancing predictive model performance are the primary objectives of feature selection. Considering that, large number of works have been done in the literature [18]. In general, these methods can be classified into three main categories: Filter, Wrapper, and Embedded.

Filter-type methods such as ReliefF [6] and MI (Mutual Information) [8] use variable ranking techniques as the main criteria for selecting variables [9]. The attributes are given a score based on a suitable ranking criterion, and attributes that are deemed unnecessary or uninformative below the threshold are removed.

Mutual Information

Let X be a variable and Y a target variable, the impact that X can have on another variable can be measured by the amount of mutual information (MI) exchanged between X and Y, such that the variable having the most mutual information with the target is the best. MI is mathematically defined as follows:

$$MI(X,Y) = \sum_{x\in X}\sum_{y\in Y} p_{(X,Y)}(x,y).log\left(\frac{p_{(X,Y)}(x,y)}{p_X(x)p_Y(y)}\right)$$

MI is zero when X and Y are statistically independent $(p_{(X,Y)}(x,y) = p_X(x)p_Y(y))$.

Relief

The impact of the variable X on the target Y can be measured in another way by measuring the degree of separation of the classes of Y by the instances of X : This is the Relief method proposed by Kira and Rendell in [6]. This method consists in finding the nearest neighbors X_{neighA} of the same class of X_i and X_{neighB} of a different class of X_i. And therefore, if the value of X_i is different from X_{neighA}, the attribute X separates two instances of the same class, which is not desirable, therefore, the weight of the attribute X will decrease. On the other hand, if X_i and X_{neighB} have different values, this means that X separates two instances of different classes, which is desirable, and therefore, the weight of X increases.

$$W_X = W_X - (X_i - X_{neighA})^2 - (X_i - X_{neighB})^2$$

On the other hand, wrapper methods such as SFS (Sequential Forward Selection) [5] and RFE-SVM (Support Vector Machine Recursive Feature Elimination) [10] use an iterative feature selection process based on prediction performance to find the best subset of features. Compared to a filter approach, this approach generally obtain better performances [7] but is significantly more expensive because it requires training and cross-validating the model for every feature subset combination.

Sequential Forward Selection

This method was proposed by [5] to select attribute subsets sequentially by taking into account the model's performance on these subsets. Let A be the set of attributes in the database and V_i be the subset of attributes of size i selected by SFS. Initially, all $V_i = \{\}$, $\forall i \in \{1, ..., |A|\}$. We calculate the prediction performance of each attribute of A and select the attribute A_j having obtained the best performance, hence $V = \{A_j\}$. Let $B = A - V_1$ to obtain V_2, we calculate the performance of the attribute subsets $V_1 \cup \{B_k\}$, $\forall i \in \{1, ..., |A| - 1\}$. The subset with the best performance will be V_2 and so on for the other sizes.

Recursive Feature Elimination of Support Vector Machines (RFE-SVM)

The RFE-SVM method was proposed by Guyon in [10]. This method like SFS selects attribute subsets depending on the prediction results of a model on these attribute subsets, whereas it initially starts with all attribute sets, uses SVM model to assign weights to attributes, and the attribute with the smallest weight will be removed from the attribute set. Let A be the set of attributes in the database and V_i be the subset of size 2 selected by RFE-SVM. Initially $V_{|A|} = A$, use

SVM to assign weights to all attributes of $V_{|A|}$, choose the attribute with the smallest weight and remove this attribute to get $V_{|A|-1}$. This will be done recursively and sequentially, from the largest size to the smallest size.

Embedded feature selection methods integrate the feature selection machine learning algorithm as part of the learning algorithm, in which classification and feature selection are performed simultaneously. The features that will contribute the most to each iteration of the model training process are carefully extracted. Compared to the wrapper strategy, the Embedded feature selection methods results in lower computational costs than wrappers. The LASSO [3] and Ridge [4] regularization approaches are common embedded methods.

LASSO (L1-Regularization)

Introduced by Robert Tibshirani in 1996 [3], LASSO is an efficient tool for regularization function selection (LASSO). The LASSO approach penalizes the number of absolute values of the machine learning algorithm parameters. Any coefficients are reduced to zero by the additional limit (regularization). During the feature selection process, features with non-zero coefficient after the removal step are selected for use in the model, while those with exactly zero coefficient are omitted.

$$Min \;\; J(W) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{k=1}^{m} |W_k|$$

Where $\hat{y}_i$ is the prediction for observation, y_i is the actual value for observation, W_k are the regression coefficients and λ is a hyperparameter that controls the intensity of the regularization.

Ridge (L2-Regularization)

The Ridge method like LASSO is a method proposed by Arthur Hoerl and Robert Kennard in [4] with the aim of avoiding over-learning regression models by assigning a penalty based on L2 regularization. Ridge's penalty like Lasso's forces the regression coefficients to be smaller, thus avoiding having too large values and reducing the risk of overfitting. Except that compared to LASSO's penalty, Ridge's penalty shrinks the coefficients to values close to zero. The loss function used by Ridge is:

$$Min \;\; J(W) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 + \lambda \sum_{k=1}^{m} (W_k)^2$$

Where $\hat{y}_i$ is the prediction for observation, y_i is the actual value for observation, W_k are the regression coefficients and λ is a hyperparameter of regularization.

In 2024 Neetha in [28] proposes a feature selection method called adaptive manta ray foraging optimization (AMRFO), in a context of brain tumor classification with the aim of selecting irrelevant features which lead to a classification error.

2.2 Association rule Feature Selection

Association rule learning is a rule-based machine learning method to discover interesting relationships between variables in large databases. It can be used to identify strong rules between attributes in the dataset.

In 2010, Chawla [12] proposed to use Association Rules Networks (ARN), where regression analysis [13] will be used to formally test the relationship between the extracted attributes after applying a minimal cut partition clustering algorithm to extract the pertinent attributes.

In 2017, Salma [11] proposes to select attributes using association rules based on constraints. The constraint on the association rules used here is that the before part of the rule must be a value of the target attribute, and the after part of the association rules must be of dimension one.

The association rule feature selection proposed by Qu in 2019 [14] will use the association rules to create a vector of attribute scores whose score for an attribute represents the highest confidence of the rules that are associated with that attribute. These association rules have the following constraints: the antecedent must be a modality of an attribute, and the consequent must be a class modality. Then, this vector will be arranged in descending order to position the attributes having a correlation with the class at the top of the list. This vector of scores will be used to select the best subset of attributes from the SFS strategy (Sequential Feature Selection). Initially, the attribute subset is empty, and an attribute from the attribute score vector is added into the attribute subset if adding it improves prediction performance.

2.3 Cost sensitive Feature Selection

The selection of cost-sensitive characteristics makes it possible to take into account costs related to the context in the selection process to improve the quality of the results.

In 2012, He and Jason [21] proposes a dynamic feature selection algorithm that sequentially chooses features based on previously selected features and their values, and stops the selection process to make a prediction according to a user-specified accuracy-cost trade-off.

Later in 2017, Liu[18] proposes to explore the class imbalance issue of data by optimizing F-measures decomposed into a series of cost-sensitive classification problems. He investigate the cost-sensitive feature selection by generating and assigning different costs to each class with rigorous theory guidance.

In 2019, Zhao [20] suggests using the $l_{2,1}$ norm to create a cost-sensitive feature selection algorithm. In order to guarantee that every feature chosen is independent, it also suggests adding an orthogonal constraint term to reduce testing expenses and classification error costs at the same time. Huang [16] presents the concept of label meaning in cost-sensitive feature selection in the same year and suggests a feature selection method that bases testing cost on label meaning.

In 2021, Long [17] suggests using neighborhood granularity and label improvement in 2021 to choose cost-sensitive features on multi-label data. He takes into account both the process cost of gathering thematic qualities like money, time, and others, as well as the comparatively varied relevance of the labels associated with a given data instance.

The same year, Pes [2] did a comparative study on cost-sensitive learning strategies for high-dimensional and imbalanced data. He validate the advantageous effect of integrating cost-sensitive learning with feature selection, particularly when highly skewed data distributions are present. Reviewing Barrera [1] work in 2023 on feature selection problems helps us better understand the 161 articles published between 2019 and 2023 (20 April 2023), emphasizing the formulation of the problem and performance measures, and proposing classifications for objective functions and evaluation metrics.

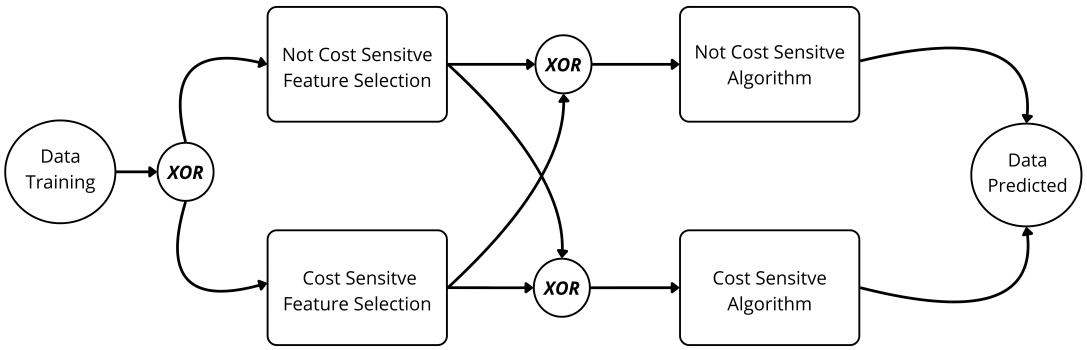
It is evident that most of these works nest cost sensitivity into feature selection methods by assigning misclassification costs depending on the context.

2.4 Financial cost in Datamining

In 2018 Metzler[22] proposes learning strategies based on cost-sensitive trees, applied in the context of highly imbalanced data. Initially, he suggested a cost-sensitive splitting criterion for decision trees that considers the transaction costs. He expands on this with a decision rule for classification with sets of trees. He then proposes a new cost-sensitive loss function by computing the gradient. Both methods have proven to be particularly relevant in the context of unbalanced data, particularly in the context of fraud detection. Experiments show that these cost-sensitive algorithms can increase retailer profits by 1.43% compared to non-cost-sensitive algorithms and that the gradient boosting approach outperforms all its competitors.

This section allowed us to present four essential ideas (an aggregation of the principle, advantages and disadvantages has been presented in the table 1) which are feature selection, cost-sensitive feature selection, feature selection based on association rules and financial cost in data mining. Using the SFS strategy on a vector of attribute scores obtained from association rules to select a better subset of attributes sensitive to a cost function proposed by Qu [14], assigning a cost to classes for each misclassification and taking into account these misclassification costs in a gradient boosting model proposed by Metzler in [22] and taking into account misclassification costs in attribute selection algorithms to have cost-sensitive attribute selection algorithms to take into account data imbalance in certain contexts, lead us to implement a cost-sensitive attribute selection method to reduce the banker's overall losses related to misclassification, based on association rules and dependent on a cost-sensitive Gradient Boosting classification model.

Fig. 1. General model representation



3 Methodology

Figure 1 presents the general model studied in this paper, as input we have training data, we will then go through one of the possibilities, that is to say: 1- Cost sensitive feature selection and cost sensitive algorithm, 2- Not Cost sensitive feature selection and not cost sensitive algorithm, 3-Cost sensitive feature selection and not cost sensitive algorithm, 4-Not Cost sensitive feature selection and cost sensitive algorithm.

As part of this paper we focus on the first possibility because prediction algorithms sensitive to finan-

Table 1. *Comparison of different concepts related to the selection of attributes sensitive to financial costs.*

Concept	Principle	Benefits	Disadvantages
Feature Selection	These algorithms reduce the number of variables in a dataset while retaining the information most relevant to the learning task.	***Dimensionality Reduction*** (Fewer attributes means less model complexity), ***Performance Improvement*** (By removing redundant or irrelevant attributes, the model becomes more efficient and accurate), ***Overfitting Prevention*** (By reducing the number of attributes, the risk of overfitting is reduced), ***Computational Time Saving*** (Less data to process speeds up the training and inference processes).	***Potential loss of information*** (Risk of removing attributes that could be useful in some complex interactions), ***Computational cost*** (Methods like wrappers can be computationally expensive), ***Data dependency*** (Biased or noisy data can mislead the selection.)
Feature Selection from Association Rules	The most important feature is the one with the greatest correlation with the target or with other features.	***Better interpretability of results***, ***generation of different feature ranking models from metrics like support, confidence*** (These models can be considered as base models to extract a better model.)	***High computational cost*** (Generating association rules from large datasets can be expensive in terms of computational time and memory, especially if the number of features or items is very high.), ***Parameter sensitivity*** (The choice of support and confidence thresholds can influence the results obtained.), ***Difficulty in processing real data***(Association rule mining algorithms work well when the data is categorical.)
Cost-Sensitive Feature Selection	These are designed to take into account not only model performance (such as accuracy), but also the cost associated with prediction errors.	***Cost optimization*** (They allow minimizing overall costs by taking into account the financial impacts of errors and data costs.), ***More realistic decision*** (In environments where the cost of errors varies considerably, these algorithms allow us to better adapt models to economic realities), ***Improved contextual performance*** (These models can be more effective in situations where raw performance is not the only criterion of success.)	***Complexity*** (Considering costs makes algorithms more complex to design and implement.), ***Reliance on cost estimates*** (The success of these methods depends heavily on accurate cost estimates, which can be difficult to obtain or model in some contexts.), ***Loss of overall accuracy*** (By focusing on minimizing costs, the overall accuracy of the model can be reduced if priorities are too biased toward financial aspects or acquisition costs.)
Financial Cost-Sensitve in Datamining	Not all errors are equally important in some contexts during model building. In the case of credential scoring, an error in not predicting a default can have a much higher cost than an error in classifying a repayable loan as non-repayable.	***Economic impact-based optimization***(This approach prioritizes errors based on their true cost.), ***Better decision-making*** (By taking costs into account, decisions are more aligned with the company's strategic objectives.), ***Flexibility*** (These models allow for adjusting error weights as needed.), ***Risk reduction*** (Minimizing costly errors can help reduce financial losses.)	***Increased complexity***(The costs of different types of errors need to be assessed and estimated, which is not always easy or intuitive.), ***Possible reduction in overall accuracy*** (By prioritizing certain errors based on costs, the overall accuracy of the model may decrease).

cial costs have been proposed to better reduce financial losses compared to algorithms not sensitive to costs,

which allows the algorithm to Cost-sensitive selection proposed in this paper to better reduce financial losses.

3.1 Decision system with misclassification cost function

In data mining and machine learning [15], decision systems with misclassification costs are an important concept and are defined as follows. A DS decision system is a 5-tuple: $DS = (U, C, D, V, f)$ where:

1. U : A finite non-empty set of objects called a universe
2. C : A non-empty finite set of condition attributes
3. D : $D = \{d\}$ is a finite set of decision attributes
4. V : $V = \{V_a\}$ is a set of values for each attribute such that $a \in C \cup D$
5. f : $f = \{f_a\}$ is an information function for each attribute $a \in C \cup D$ ($f_a : U \times (C \cup D) \Rightarrow V_a$)

For example, table 2 presents a decision system where $U = \{1, 2, 3, 4, 5, 6\}$, $C = \{City, Occupation, Age, Amount, Interest\ rate\}$ and $D = \{Class\}$.

Table 2. *Example Dataset*

ID	City	Occupation	Age	Amount	Interest rate	Class
1	Yaoundé	0	30	500000	2	PAID
2	Yaoundé	1	45	120500	1.5	PAID
3	Douala	3	40	5000000	3	UNPAID
4	Yaoundé	0	25	500000	2	UNPAID
5	Yaoundé	0	42	3000000	1.5	PAID
6	Yaoundé	0	50	800000	2	PAID

A decision system with misclassification cost (MC-DS) can then be defined as a 6-tuple: MC-DS = (U, C, D, V, f, mc) where :

1. U, C, D, V, f have the same meanings as in the previous definition.
2. mc is a misclassification cost function ($C \cup D \Rightarrow \mathbf{R}^+ \cup \{0\}$)

In classification problems with two classes $y_i \in \{0, 1\}$, the goal is to learn or predict which class $c_i \in \{0, 1\}$ a given example i belongs according to its k characteristics $X_i = [x_i^1, x_i^2, ..., x_i^k]$. In this context, classification costs can be represented using a cost matrix 2×2, which introduces the costs associated with two types of correct classification, true positives (C_{TP_i}) , true negatives (C_{TN_i}), and two types of classification errors, false positives (C_{FP_i}), false negatives (C_{FN_i}).

Let S be a set of N examples i, $N = |S|$, where each example is represented by the augmented feature vector $X_i^a = [X_i, mc_i]$ and labeled using the class label $y_i \in \{0, 1\}$. According to the banker :

1. For a bad borrower announced as good, we lose the amount granted or at least part of it.
2. For a good borrower announced as bad, there is a deficit (sf).

Assuming this, we define a misclassification cost function cst as :

$$mc_i = \begin{cases} C_{FP_i} = X_i[amt] \times X_i[lgd] & if \quad y_i = 0 \\ C_{FN_i} = sf = X_i[amt] \times X_i[rt] \times X_i[d] & else \end{cases} \tag{1}$$

Where $X_i[amt]$, $X_i[rt]$, $X_i[d]$, $X_i[lgd]$ represent the amount, rate, duration, and loss given default of the given loan, respectively. The cost of a classification error for a set S is :

$$mc(S) = \sum_{i}^{|S|} mc_i \tag{2}$$

A classifier f that generates the predicted label c_i for each element i is trained using the set S. The cost of using the classifier f on S is calculated by :

$$\begin{aligned} Cost(f(S)) = \sum_{i=1}^{N} [y_i(c_i C_{TP_i} + (1 - c_i)C_{FN_i}) + \\ (1 - y_i)(c_i C_{FP_i} + (1 - c_i)C_{TN_i})] \end{aligned} \tag{3}$$

3.2 Cost Sensitive Gradient Boosting

Gradient Boosting Unlike the well-known Adaboost algorithm [19], gradient boosting performs optimization in the function space rather than in the parameter space. At each iteration, a weak learner f_t is learned using the residuals (or errors) obtained by the linear combination of the previous models. The linear combination F_t at time t is defined as follows:

$$F_t = F_{t-1} + \alpha f_t \tag{4}$$

where F_{t-1} is the linear combination of the first $t - 1$ models and α_t is the weight given to the t^{th} weak learner. The weak learners are trained on the residuals $r_i = y_i - F_{t-1}(x_i)$ of the current model. These residuals are given by the negative gradient $-g_t$, of the used loss function L with respect to the current prediction $F_{t-1}(x_i)$:

$$r_i = -g_t = -\left[\frac{\partial L(y, F_{t-1}(x_i))}{\partial F_{t-1}(x_i)}\right] \tag{5}$$

Once the residuals r_i are computed, the following optimization problem is solved :

$$(f_t, \alpha_t) = \arg\min_{\alpha, f} \sum_{i=1}^{m} (r_i - \alpha f(x_i))^2 \tag{6}$$

Cost sensitive loss for gradient boosting Metzler in [22] suggests using the cost function L which derives from the equation (3). For $\hat{F}_i = F(x_i)$ we have :

$$L(y|\hat{F}_i) = \frac{1}{m} \sum_{i=1}^{m} (1 - s_i) y_i e^{-\hat{F}_i} + (1 - y_i) s_i e^{\hat{F}_i} \tag{7}$$

To use it in a gradient boosting algorithm, it remains to compute the first and second order derivative of L

for each instance i with respect to $\hat{F}_i$. They are given by :

$$\frac{\partial L}{\partial \hat{F}_i} = \xi_i[-(1-s_i)y_i e^{-\hat{F}_i} + (1-y_i)s_i e^{\hat{F}_i}] \quad (8)$$

and

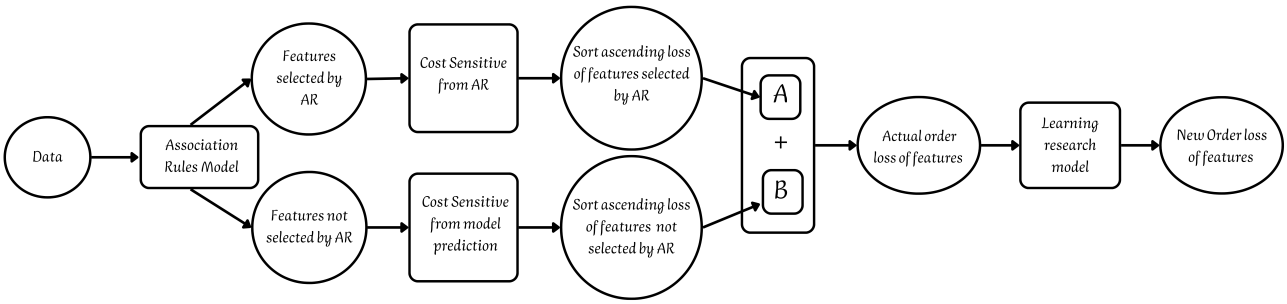
$$\frac{\partial L}{\partial \hat{F}_i^2} = \xi_i[(1-s_i)y_i e^{-\hat{F}_i} + (1-y_i)s_i e^{\hat{F}_i}] \quad (9)$$

where $\xi_i = C_{TN_i} - C_{FP_i} + C_{TP_i} - C_{FN_i}$,

$$s_i = \frac{C_{TN_i} - C_{FP_i}}{C_{TP_i} - C_{FN_i} + C_{TN_i} - C_{FP_i}}$$

and $\forall i \in \{1, .., m\}$ $C_{FN_i} < 0$, $C_{FP_i} < 0$, $C_{TP_i} > 0$ and $C_{TN_i} > 0$.

Fig. 2. Specific model representation



Algorithm 1: L_2 Mining algorithm

Input: supp_min : support minimal seuil
Output: Conf : confiance des item sets fréquents de taille 2
Data: D : Ensemble de données

1 *init* P_2, *supp_min*
2 $n \leftarrow count(P_2)$
3 **for** $i < n$ **do**
4 $\quad supp_{P_{2_i}} \leftarrow support(P_{2_i})$ $\quad //P_{2_i} = (X_i, Y)$
5 $\quad$ **if** $supp_{P_{2_i}} > supp\_min$ **then**
6 $\quad\quad L_2.add(P_{2_i})$
7 $\quad\quad conf_{2_i} \leftarrow \frac{count(L_{2_i})}{count(X_i)}$
8 **return** L_2.Conf

3.3 Cost Sensitive Association Rules Feature Selection more Large (ARFSL-CS)

We propose the specific model shown in Figure 2, which uses association rules and cost sensitivity [5] to provide a financial loss-sensitive order of the features, and then uses a learning research model to derive a new order of attributes, such that by extracting the subset of attributes based on their sizes using the k-best strategy, we will only have better subsets of attributes. The learning search model considers the order of attributes as a model. Adjusts this during the search and will provide a new model that represents a new order of attributes.

The ARFSL-CS proposed in this paper and presented in algorithm 2 must first calculate the confidence of the set of frequent item sets of size 2 denoted L_2 between the attribute modalities and the category denoted $\{V = v, C = c\}$ from L_2 Mining algorithm 1 of Qu [14] to the lines 2 and 3 in algorithm 2.

Then, on line 4, we group together all the item sets whose attribute is V, then we associate with each association rule a cost equal to the confidence of this rule multiplied by the loss of rules which will be calculated using the cost function $mc(S)$ defined in the previous subsection. The loss of the rule will represent the sum of losses for which the rule is verified. We then select the maximum cost of this group of item set such that the rule is $\{V = v\} \Rightarrow \{C = c\}$, this score will represent the score of the attribute V. We will have a loss attribute vector called the selected attribute vector V_{select} by defining minimum support and minimum confidence. The confidence of a rule is used here to measure the correlation between attributes and categories.

The multiplication of loss and confidence in the ARFSL-CS algorithm is due to the fact that assuming that we have two association rules $R_1 : \{A = a\} \Rightarrow \{C = c_1\}$ and $R_2 : \{B = b\} \Rightarrow \{C = c_2\}$ with the same loss of p calculated from mc but a respective confidence of 1.0 and 0.8. By multiplying $p \times 1.0$ there is no change; on the other hand, by making $p \times 0.8$ the loss decreases. This is explained by the fact that if the attribute is strongly correlated to a class, it will not distinguish the other classes at best hence its loss does not decrease, on the other hand if the attribute is not strongly correlated to a class, it can have a correlation to another class distinguish at least two other classes hence its loss decreases.

[5] Cost sensitivity from association rules and from the prediction model.

Due to minimal support and minimal trust, some rules will be removed, which will result in some attributes being removed. The attributes thus deleted will be used to form the forgotten attribute vector V_{forget} to lines 5 and 6. The forgotten attribute vector V_{forget} will be constructed by extracting attributes that belong to the starting attribute set, but do not belong to the attribute set of the selected attribute vector. Then, calculate the financial loss of each forgotten attribute using the Ft-CS learning model. Then, the selected V_{select} and forgotten V_{forget} attribute vectors are arranged in ascending order on lines 7 and 8 and concatenated on line 9 to obtain the attribute score vector which will be passed to the GLRFSL-CS algorithm 3. We can then search for the best subsets of attributes based on the size of the subset using the GLRFSL-CS (Cost Sensitive Global and local Research Feature Subset by Learning scored vector) search algorithm proposed in this paper in algorithm 3. The idea of this algorithm is to leave the current attribute order to provide a new attribute order in such a way that it allows us to extract subsets of attributes that reduce the banking loss. $\eta_{cost}(f, D, GB-CS)$ returns the banker's loss for a training set D, on characteristics f using a model sensitive to financial costs $GB-CS$.

Algorithm 2: Cost Sensitive Association Rules Feature Selection more Large (ARFSL-CS)

Input: sup_min : minimum support threshold, conf_min : minimum confidence threshold
Output: set_opt_loss : Best cost sensible features subset
Data: D : Dataset

1 *Init* $GB-CS$ // *Cost Sensitive Grandient Boosting*
2 $T_2 \leftarrow Apriori(sup\_min, D)$
3 $R_2 \leftarrow Association\_Rules(T_2, conf\_min)$
4 $V_{select} \leftarrow MAX(R_2 \times mc(R_2))$
5 $V_{forget} \leftarrow Get\_feature\_forget(D, V_{select})$
6 $V_{forget} \leftarrow Get\_Feature\_Loss(GB-CS, D, V_{forget})$
7 $V_{select} \leftarrow Sort\_ascending(V_{select})$
8 $V_{forget} \leftarrow Sort\_ascending(V_{forget})$
9 $V_{loss} \leftarrow V_{select} + V_{forget}$
10 $d \leftarrow divide\_lenght$
11 $set\_opt\_loss \leftarrow GLRFSL-CS(V_{loss}, GB-CS, d)$
12 **return** *set_opt_loss*

Firstly, we add the characteristic V^i_{loss} to the vectors z and w in lines 4 and 5, then we adjust $opt[|z|]$ and $opt[|w|]$ locally on lines 6 and 7 if the loss on the subset of attributes z and w is, respectively, lower than the loss on the subset of attributes $opt[|z|]$ and $opt[|w|]$. Then put into f the attribute subset with the smallest loss (the line 8) and compare f to the current best attribute subset f_{cost}, adjust V_{loss} from f if f has a loss strictly lower than that of f_{cost}.

Then, on line 14, a step of the SBS (Sequential Backward Selection) strategy is used to have the best subset of reduced size among that selected by the SFS strategy. This idea was taken from the Sequential Floating Forward Selection (SFFS) algorithm of [23]. During this process, an adjustment to the attribute loss vector is made each time a new subset of attributes with minimal financial loss is discovered.

Algorithm 3: Cost Sensitive Global and Local Research of Feature Subset by Learning (GLRFSL-CS)

Input: V_{loss} : Scores vector of feature, GB-CS : Learning model, d : step
Output: opt : Best cost sensible features subset
Data: D : Dataset

1 $\beta \leftarrow |V_{loss}|$
2 $i \leftarrow 1$
3 **while** $valid == False$ **do**
4 $\quad z.add(V^i_{loss_d})$
5 $\quad w.add(V^i_{loss_d})$
6 $\quad opt[|z|] \leftarrow Min(\eta_{cost}(opt[|z|], D, GB-CS), \eta_{cost}(l, D, GB-CS))$
7 $\quad opt[|w|] \leftarrow Min(\eta_{cost}(opt[|w|], D, GB-CS), \eta_{cost}(g, D, GB-CS))$
8 $\quad f \leftarrow Min(\eta_{cost}(z, D, GB-CS), \eta_{cost}(w, D, GB-CS))$
9 $\quad$ **if** $\eta_{cost}(f, D, GB-CS) < \eta_{cost}(f_{cost}, D, GB-CS)$ **then**
10 $\quad\quad opt[|z|], f_{cost}, g \leftarrow f, f, f$
11 $\quad\quad$ *update* f_{cost} *in* V_{loss_d} *with ascending order*
12 $\quad$ **else**
13 $\quad\quad g.pop()$
14 $\quad$ *SBS step on* $opt[m]$ *and update* V_{loss} *if needed*
15 $\quad$ *Test* $[opt[m], V^i_{loss_d}]$ *and update* V_{loss} *if needed*
16 $\quad$ **if** $i > \beta$ **then**
17 $\quad\quad valid \leftarrow True$
18 $\quad i \leftarrow i + d$
19 **return** *opt*

Test the new attribute subset at the line 15 in order to select better attribute subsets by concatenating $opt[m]$ to V^i_{loss}, during this process, adjustments to the score vector V_{loss} can be done in case of reduction of losses. ARFSL-CS can be used with all learning algorithms, whether cost-sensitive or not, for the purpose of selecting attributes that minimize losses. The mc of the equation (2) function can be adjusted depending on the context in which we find ourselves.

Table 3. *Example of Data set*

A	B	C	D	E	F	Target
y	5M	30	2	2	0	1
y	3M	45	2	1.5	1	1
d	500k	40	1	3	3	0
y	500k	25	1	2	0	0
y	120k	42	1	1.5	0	1
y	800k	50	2	2	0	1

Table 4. *Rules obtained with a support of* 0.33 *and a confidence of* 0.66.

Before	After	Conf	$Loss_1$	$Loss_2$
A=y	Target=1	0.8	26983	21586
D=2	Target=1	1.0	26833	26833
D=1	Target=0	0.66	2083	1374
E=2	Target=1	0.66	20166	13309
E=1.5	Target=1	1.0	7650	7650
F=0	Target=1	0.75	19483	14612

Table 5. *Rules obtained with a support of* 0.33 *and a confidence of* 0.8.

Before	After	Conf	$Loss_1$	$Loss_2$
A=y	Target=1	0.8	26983	21586
D=2	Target=1	1.0	26833	26833
E=1.5	Target=1	1.0	7650	7650

3.4 Application example

The purpose of this example is to highlight the cost sensitivity in the ARFSL-CS algorithm presented in 2 using association rules. Table 3 presents an example dataset. In this table, the features A="City", B="Amount", C="Age", D="Duration", E="Interest rate", F="Occupation".

Table 4 presents the association rules obtained from the support thresholds equal to 0.33 and the confidence equal to 0.66. In this table, the attribute **Before** represents the antecedent of the rule, **After** represents the consequent and **Conf** the confidence of an association rule. In addition, the attribute $Loss_{2_i} = Loss_{1_i} \times Conf_i$ represents the loss of confidence in a rule and $Loss_1$ (calculated using mc) the loss to be realized for this rule.

As explained in the previous subsection, the rules whose antecedent is an attribute modality and whose consequent is a modality of the target attribute are extracted. Subsequently, the rules whose antecedents are associated with the same attribute are grouped together (the colored boxes), and we assign the greatest confidence loss of each group of rules as the score of the attribute associated with this group of rules.

In table 4, for example, a group of rules is the one in red, the attribute associated with this group is the attribute D which will have as score $max(26833, 1374) = 26833$. After assigning a score to each attribute, we will obtain the loss vector of the attributes selected by the association rules by arranging these attributes ($\{A, D, E, F\}$) in ascending order.

In the table 4 the set of attributes that have not been selected by the association rules (second phase) is the set $\{B, C\}$, and the scores of these attributes B and C will be obtained by using the data related to each attribute to train a prediction model M_B and M_C, and therefore, the losses of each model M_B and M_C will represent the scores of attributes A and B respectively. After assigning the scores of attributes B and C, we obtain the attribute vector forgotten by the association rules by ranking these attributes in an increasing manner according to their scores.

The concatenation of the vector of attributes selected by the association rules and that of the attributes forgotten by the association rules will be passed to the search algorithm based on SFS and SBS. The same process will be performed for the rules in the table 5 with a set of selected attributes $\{A, D, D\}$ and a set of forgotten attributes $\{B, C, F\}$.

4 Experiments

In this part, we compare the ARFSL-CS algorithm to the LASSO, Ridge, ReliefF and MI algorithms with the aim of observing the impact of taking into account cost sensitivity in the selection of attributes.

The data used in the experiment are: LDB, German Credit Data [24] and Credit Risk [6]. German and Australian datasets come from UCI Machine Learning Repository, LDB is a dataset from a local bank. This experiment was carried out on a 16 GB RAM using the Sklearn Python tool for learning models.

We used the xgboost python tool from sklearn, which implements grandient bossting, for the construction of prediction models without cost sensitivity and with cost sensitivity [7].

4.1 Dataset

Table 6. *Dataset*

Dataset	Instances	Attributes	Class
LDB	28952	12	2
German	1000	20	2
Credit Risk	28638	12	2

We start by applying these algorithms on 3 different datasets as presented at the table 6 :

- LDB : This dataset of 28,952 instances gathers information from various individuals or individuals registered in a local bank where each individual has 17 attributes and one target attribute.
- German Credit Data [24] : This dataset classifies people described by a set of attributes as having good or bad credit risk. It has 20 attributes and one target attribute.
- Credit Risk Dataset [8] : This dataset contains 28,638 instances, 11 attributes and a target attribute.

4.2 Evaluation Metrics

We use the following metrics to compare the results of different models :

- Gain : Represents the Gain in losses of using attribute selection algorithms on a prediction model :
$$Gain(f, F, X) = \frac{f(X) - F(f, X)}{f(X)}$$
or f is model prediction, F is feature selection algorithm and X est Data set. The greater its value, the more the attribute selection algorithm reduces losses.
- Rank : Represents the rank of the selection algorithm in terms of gain loss[9]. The smaller the value, the better the selection algorithm has.
- NFS : Number of features selected. The smaller its value, the more the attribute selection algorithm selects fewer attributes.

[6] `https://www.kaggle.com/datasets/laotse/credit-risk-dataset/`
[7] `https://github.com/dmlc/xgboost`
[8] `https://www.kaggle.com/datasets/laotse/credit-risk-dataset/`
[9] If both ranks 1, then they rank all 1.5 [20]

We will then explore the results in terms of accuracy, precision, and recall of the cost-sensitive algorithm ARFSL-CS which obtains the banker's losses from $mc(\text{CST})(2)$, in order to observe their deviations compared to the base algorithm without feature selection. In order to obtain statistically significant experimental results, a 5-fold cross-validation experiment was used.

4.3 Description of results and observations

In this subsection, we describe and comment on the results obtained from the experiments on credit scoring databases:

A - Description

GB represents the gradient boosting model. $GB-L$, $GB-LR$ and $GB-H$ respectively represent the gradient boosting model with the objective functions *binary logistic*, *binary logitraw* and *binary hinge*[10]. We also have $GB-CS$ which represents the gradient boosting model with the cost-sensitive objective function as proposed by Metzler in [22].

The tables 10, 11 and 12 represent the results in terms of losses (Cst), accuracy (Acc), precision (Prec) and recall (Rec), feature selection algorithms ARFSL-CS, MI, ReliefF, Ridge and LASSO on gradient boosting GB models. The *Base* algorithm returns the results of prediction models without attribute selection.

The tables 7, 8 and 9 represent the results in terms of percentage of gains, rank of algorithms in terms of gains and the number of features selected (NFS) by the ARFSL-CS, MI, ReliefF, Ridge and LASSO algorithms. The gain in loss is obtained by calculating the difference in loss when there is selection of attributes and the loss when there is no selection of attributes (**Base**).

Table 7. *Gain, Rank, and number of features selected results on LDB Dataset*

Model	Metric	MI	ReliefF	LASSO	Ridge	ARFSL-CS
GB-L	Gain	1.42%	1.30%	1.30%	0%	**1.93%**
	Rank	2	3.5	3.5	5	**1**
	NFS	**8**	11	11	12	**8**
GB-LR	Gain	1.33%	2.65%	2.65%	0.11%	**8.19%**
	Rank	4	2.5	2.5	5	**1**
	NFS	8	11	11	11	**7**
GB-H	Gain	1.49%	0.59%	0.59%	2.31%	**15.01%**
	Rank	3	4.5	4.5	2	**1**
	NFS	**9**	11	11	**9**	**9**
GB-CS	Gain	8.33%	0%	0.30%	0.57%	**21.05%**
	Rank	2	5	4	3	**1**
	NFS	**5**	12	10	10	6
Means	Gain	3.14%	1.135%	1.21%	0.74%	**11.54%**
	Rank	2	4	3	5	**1**
	NFS	**7.5**	11.25	10.75	10.5	**7.5**

Table 8. *Gain, Rank, and number of features selected results on German Dataset*

Model	Metric	MI	ReliefF	LASSO	Ridge	ARFSL-CS
GB-L	Gain	14.28%	0%	8.65%	9.49%	**18.70%**
	Rank	2	5	4	3	**1**
	NFS	18	20	19	16	**15**
GB-LR	Gain	15.05%	0%	0%	3.47%	**24.18%**
	Rank	2	4.5	4.5	3	**1**
	NFS	15	20	20	16	**3**
GB-H	Gain	6.97%	0%	0%	8.47%	**20.19%**
	Rank	3	4.5	4.5	2	**1**
	NFS	**15**	20	20	16	**15**
GB-CS	Gain	69.53%	69.53%	58.35%	69.53%	**69.94%**
	Rank	3	3	5	3	**1**
	NFS	**1**	**1**	**1**	**1**	3
Means	Gain	26.45%	17.38%	16.75%	22.74%	**33.25%**
	Rank	2	4	5	3	**1**
	NFS	12.25	15.25	15	12.25	**9**

Table 9. *Gain, Rank, and number of features selected results on Credit Risk Dataset*

Model	Metric	MI	ReliefF	LASSO	Ridge	ARFSL-CS
GB-L	Gain	0%	3.17%	0%	5.92%	**15.43%**
	Rank	4.5	3	4.5	2	**1**
	NFS	11	10	11	10	**9**
GB-LR	Gain	9.17%	**10.86%**	0%	7.35%	9.17%
	Rank	2.5	**1**	5	2	2.5
	NFS	**10**	**10**	11	**10**	**10**
GB-H	Gain	0.59%	0%	0%	3.79%	**11.90%**
	Rank	3	4.5	4.5	2	**1**
	NFS	**9**	11	11	10	**9**
GB-CS	Gain	**8.76%**	4.29%	0%	2.95%	**8.76%**
	Rank	**1.5**	3	5	4	**1.5**
	NFS	**10**	**10**	11	**10**	**10**
Means	Gain	4.63%	4.58%	0%	5.0025%	**11.31%**
	Rank	3	4	5	2	**1**
	NFS	10	10.25	11	10	**9.5**

B - Best rank and gain in loss

The results of tables 7, 8 and 9 show that ARFSL-CS is the best selection algorithm in terms of gain in losses. In the LDB and German datasets, ARFSL-CS obtained a rank of 1 on all gradient boosting prediction models as presented in figures 7 and 8 which shows that ARFSL-CS mostly obtains better gains in loss compared to other state-of-the-art attribute selection algorithms such as MI, ReliefF, LASSO, Ridge. In contrast, in the results of the Credit Risk dataset presented in table 9, ARFSL-CS obtains a rank of 1 on all gradient boosting prediction models except the gradient boosting model with a binary *logitraw* objective function denoted GB-LR, where the ReliefF algorithm was better with a gain in losses of 10.86%, which shows that in this specific case, the correlation selection strategy compared to the ReliefF target variable is better than the SFS strategy implemented by ARFSL-CS.

ARFSL-CS has a higher percentage of average gain in all datasets, as presented in Tables 7, 8 and 9, which

[10] These models represent the models not sensitive to financial costs

shows that on average, ARFSL-CS generated better gains in terms of bank losses. These results also show that on average, ARFSL-CS scores better compared to other state-of-the-art feature selection algorithms. We also note that MI obtained the second rank in terms of loss gain on the LDB and German data sets as presented in tables 7 and 8.

In the results of table 7, from the LDB dataset, the ARFSL-CS model makes a banking loss gain of 21.05% with the cost-sensitive gradient-boosting prediction model noted GB-CS, which corresponds to $47,769,325$ XAF (forty-seven million XAF). For the German and Credit Risk datasets, the results in tables 8 and 9 show that ARFSL-CS made a respective bank loss gain of 69.94% and 15.43% using the GB-CS and GB-L models respectively. These results represent the best gains obtained on the three datasets LDB, German and Credit Risk, they were obtained from ARFSL-CS and two of these results are obtained using the GB-CS model (on LDB and German) which is a model sensitive to financial costs. These results also show that using a model without cost sensitivity could be interesting in reducing losses. These results do not include the comparison aspect between cost-sensitive and noncost-sensitive models in the context of minimizing bank losses overall and will be discussed in the subsection **Prediction results**.

C - Number of features selected

Concerning the number of selected attributes, the results presented in the tables 7, 8 and 9, also show that ARFSL- CS obtained the best average number of selected attributes compared to other algorithms in the LDB, German and Credit Risk datasets, showing that in addition to having obtained better gains in bank losses, ARFSL -CS has a good ability to reduce the size of attributes. We also note that MI tied with ARSFSL-CS in terms of average of selected attributes on the LDB dataset.

In general, the results presented in figure 3 show that ARFSL-CS obtains the best gain in banking losses, the best rank among the state-of-the-art algorithms presented in this paper and the best in terms of the number of selected attributes. These results show that ARFSL-CS can select attributes that make it possible to properly classify data instances with high loss costs with a limited number of attributes.

Table 10. *Prediction results on LDB dataset*

Model	Metric	Base	MI	ReliefF	LASSO	Ridge	ARFSL-CS
GB-L	CST	231336549	228051124	228324888	228324888	231336549	**226871337**
	Acc	0.9880	0.9878	0.9879	0.9879	0.9880	**0.9882**
	Prec	0.9879	0.9878	0.9878	0.9878	0.9879	**0.9882**
	Rec	0.9880	0.9878	0.9879	0.9879	0.9880	**0.9882**
GB-LR	CST	222894760	219923868	216982374	216982374	222641461	**204638331**
	Acc	0.9871	0.9871	**0.9876**	**0.9876**	0.9871	0.9851
	Prec	0.9871	0.9871	**0.9877**	**0.9877**	0.9871	0.9852
	Rec	0.9871	0.9871	**0.9876**	**0.9876**	0.9871	0.9851
GB-H	CST	237195721	233645966	235784705	235784705	231699831	**201584553**
	Acc	**0.9884**	0.9877	0.9883	0.9883	0.9882	0.9875
	Prec	0.9884	0.9877	**0.9883**	**0.9883**	0.9882	0.9875
	Rec	**0.9884**	0.9877	0.9883	0.9883	0.9882	0.9875
GB-CS	CST	226856304	207943963	226856304	226162270	225543145	***179086979***
	Acc	0.9804	0.9805	0.9804	0.9793	0.9797	***0.9809***
	Prec	0.9812	0.9814	0.9812	0.9803	0.9806	***0.9818***
	Rec	0.9804	0.9805	0.9804	0.9793	0.9797	***0.9809***

D - Prediction results

In terms of prediction results of the selection algorithms in tables 10, 11 and 12, we observe that ARFSL-CS is the only algorithm among all attribute selection algorithms cited in this paper that always reduces the banking loss, regardless of the decision tree model in which it is applied.

These results also show that the attribute selection algorithms ARFSL-CS, MI, ReliefF, Ridge and LASSO very often make it possible to reduce losses by keeping prediction results in terms of accuracy, precision and recall more or less close to the basic results which shows that the use of a selection algorithm such as ARFSL-CS makes it possible to always reduce losses without degrading the prediction performance of the models in terms of accuracy, precision and recall.

The prediction results of the LDB, German, and Credit Risk datasets presented in tables 10, 11, and 12 show that in terms of bank loss costs, the ARFSL-CS algorithm manages to obtain the smallest bank loss cost on all three datasets compared to MI, ReliefF, Lasso and Ridge algorithms. In the LDB data set we obtained $179,086,979$ XAF from the $GB-CS$ model, in the German data set we got $23,409$ from

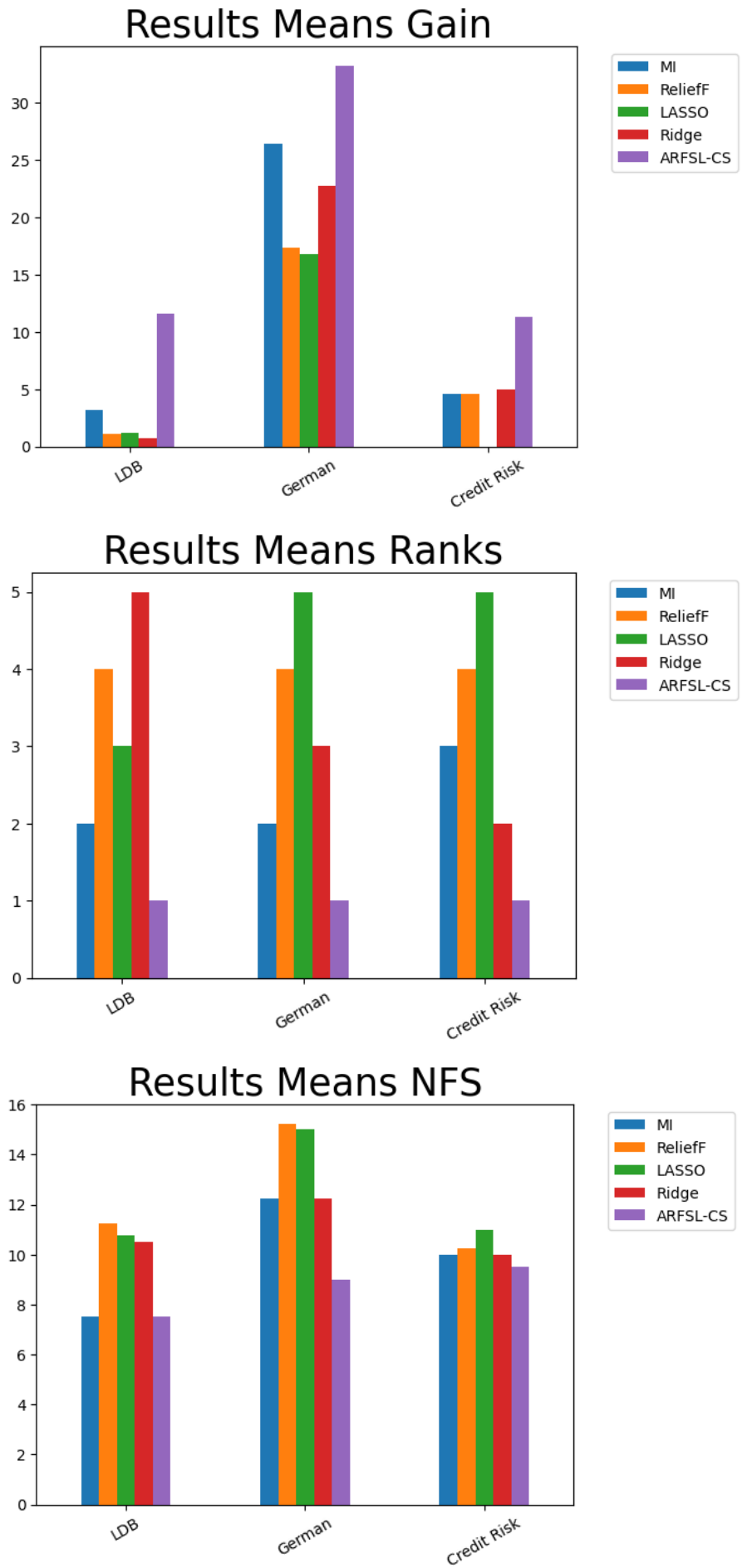


Fig. 3. Results in terms of average gain in losses, average ranks in gain and average number of selected features.

the $GB - CS$ model, and in the credit risk data set we got $302,927$ from the $GB - CS$ model. These results confirm the hypothesis that the combination of a cost-sensitive attribute selection algorithm and a cost-sensitive prediction algorithm would give very interesting results.

Table 11. *Prediction results on German dataset*

Model	Metric	Base	MI	ReliefF	LASSO	Ridge	ARFSL-CS
GB-L	CST	117631	100823	117631	107453	106460	**95631**
	Acc	0.748	0.758	0.748	0.735	0.744	**0.765**
	Prec	0.7373	0.7485	0.7373	0.7281	0.7368	**0.7618**
	Rec	0.748	0.758	0.748	0.735	0.744	**0.765**
GB-LR	CST	84924	72138	84924	84924	81977	**64381**
	Acc	**0.748**	0.756	0.74	0.74	**0.748**	0.675
	Prec	0.7445	0.7287	0.7654	0.7445	**0.7560**	0.7395
	Rec	**0.748**	0.756	0.7445	0.74	**0.748**	0.675
GB-H	CST	109625	101978	109625	109625	100329	**87483**
	Acc	0.738	0.735	0.738	0.738	**0.75**	0.73
	Prec	0.7356	0.7340	0.7356	0.7356	**0.7471**	0.7330
	Rec	0.738	0.735	0.738	0.738	**0.75**	0.73
GB-CS	CST	77898	23728	23728	32437	23728	***23409***
	Acc	**0.688**	0.3	0.3	0.317	0.3	*0.34*
	Prec	0.7459	0.0916	0.0916	0.5955	0.0916	***0.7756***
	Rec	**0.688**	0.3	0.3	0.317	0.3	*0.34*

Table 12. *Prediction results on Credit Risk dataset*

Model	Metric	Base	MI	ReliefF	LASSO	Ridge	ARFSL-CS
GB-L	CST	552191	552191	534673	552191	519482	**466947**
	Acc	0.9347	0.9347	0.9353	0.9347	**0.9357**	0.9322
	Prec	0.9362	0.9362	0.9371	0.9362	0.9375	**0.9348**
	Rec	**0.9347**	**0.9347**	0.9353	**0.9347**	0.9357	0.9322
GB-LR	CST	384433	349145	**342659**	384433	356154	349145
	Acc	0.9342	0.9343	0.9349	0.9342	**0.9354**	0.9343
	Prec	0.9374	0.9374	0.9382	0.9374	**0.9386**	0.9374
	Rec	0.9342	0.9343	0.9349	0.9342	**0.9354**	0.9343
GB-H	CST	558141	554815	558141	558141	536971	**491721**
	Acc	0.9342	0.9319	0.9342	0.9342	**0.9345**	0.9301
	Prec	0.9361	0.9335	0.9361	0.9361	**0.9365**	0.9324
	Rec	0.9342	0.9319	0.9342	0.9342	**0.9345**	0.9301
GB-CS	CST	332012	***302927***	317767	332012	322192	***302927***
	Acc	0.9112	*0.9137*	**0.9177**	0.9112	0.9129	*0.9137*
	Prec	0.9193	*0.9218*	**0.9249**	0.9193	0.9209	*0.9218*
	Rec	0.9112	*0.9137*	**0.9177**	0.9112	0.9129	*0.9137*

We also observe that, in the prediction results of the LDB and German datasets presented in tables 10 and 11, the results obtained by ARFSL-CS from the $GB - CS$[11] and $GB - L$[12] models are better in costs, accuracy, precision and recall compared to other attribute selection algorithms and even compared to the base prediction on this same model, which shows that ARFSL-CS can provide not only a subset of attributes that minimize banking losses, but also improves usual prediction results such as accuracy, precision, and recall. We also observe this in the German dataset, in the table 11 with the $GB - CS$ model, but the precision, prediction, and recall results are not always better than those of $Base$, but these results are best compared to the results of other selection algorithms.

[11] Only LDB dataset for $GB - CS$

[12] LDB and German datasets for $GB - L$

5 Conclusion

The objective of our work was to increase the overall profit of the bank by selecting features that reduce bank losses.The idea is to select a subset of attributes that will reduce bank losses during the process of predicting bank risk based on association rules. We propose to use the association rules and the prediction performance of the model to provide a vector of scores which will be used as a base model to provide a new model which allows extracting subsets of attributes which better reduce the banker's losses. The results show that on average ARFSL-CS is better in terms of reduction of bank losses, rank in gain of losses, number of attributes selection compared to state-of-the-art algorithms such as MI, ReliefF, LASSO, Ridge. It provides prediction results more or less close to results without attribute selection on recall, accuracy, and precision metrics. We observe that some state-of-the-art algorithms rank first in some cases, even though they are not cost sensitive. This is explained by the fact that these algorithms, by selecting attributes in order to improve precision, recall, or accuracy, could reduce losses because bank losses are due to bank classification errors. We also noticed that the classifier that produce better results with ARFSL-CS are cost sensitive classifiers. In future work, we will study the impact of taking cost sensitivity into account on the explainability properties of black box models.

References

1. Barrera-García J, Cisternas-Caneo F, Crawford B, Gómez Sánchez M, Soto R. Feature Selection Problem and Metaheuristics: A Systematic Literature Review about Its Formulation, Evaluation and Applications. Biomimetics (Basel). 2023;9(1):9. Published 2023 Dec 25. https://doi.org/10.3390/biomimetics9010009
2. Pes B, Lai G. Cost-sensitive learning strategies for high-dimensional and imbalanced data: a comparative study. PeerJ Comput Sci. 2021 Dec 24;7:e832. https://doi.org/10.7717/peerj-cs.832 PMID: 35036539; PMCID: PMC8725666.
3. Tibshirani, R. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267-288. (1996).
4. Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1), 55-67.
5. Raman, B., & Ioerger, T. R. Instance-based filter for feature selection. Journal of Machine Learning Research, 1(3), 1-23. (2002).
6. Kononenko, I. Estimating attributes: Analysis and extensions of RELIEF. In European conference on machine learning (pp. 171-182). Springer, Berlin, Heidelberg. (1994, April).
7. Kohavi, R. and John, G. H. (1997). Wrappers for feature subset selection. Artificial intelligence, 97(1-2) :273–324.
8. Battiti, R. Using mutual information for selecting features in supervised neural net learning. IEEE Transactions on neural networks, 5(4), 537-550. (1994).

9. Chandrashekar, G. and Sahin, F. (2014). A survey on feature selection methods. Computers & Electrical Engineering, 40(1) :16–28.
10. Guyon, I., Weston, J., Barnhill, S., & Vapnik, V. Gene selection for cancer classification using support vector machines. Machine learning, 46(1), 389-422. (2002)
11. Salma, M. U. et al. (2017). Reducing the feature space using constraint-governed association rule mining. Journal of Intelligent Systems, 26(1) :139–152.
12. Chawla, S. (2010). Feature selection, association rules network and theory building. In Feature Selection in Data Mining, pages 14–21. PMLR.
13. Pandey, G., Chawla, S., Poon, S., Arunasalam, B., and Davis, J. G. (2009). Association rules network : Definition and applications. Statistical Analysis and Data Mining : The ASA Data Science Journal, 1(4) :260–279.
14. Qu, Y., Fang, Y., and Yan, F. (2019). Feature selection algorithm based on association rules. In Journal of Physics : Conference Series, volume 1168, page 052012. IOP Publishing.
15. Zhao, H., Li, X.: A cost sensitive decision tree algorithm based on weighted class distribution with batch deleting attribute mechanism. Information Sciences 378, 303–316 (2017). https://doi.org/10.1016/j.ins.2016.09.054
16. Huang, Jintao, et al. : "Cost-Sensitive Feature Selection Based on Label Significance and Positive Region." 2019 International Conference on Machine Learning and Cybernetics (ICMLC). IEEE, (2019)
17. Long, Xuandong, et al. "Cost-sensitive feature selection on multi-label data via neighborhood granularity and label enhancement." Applied Intelligence 51 (2021): 2210-2232.
18. Liu, Meng, et al. "Cost-sensitive feature selection by optimizing F-measures." IEEE Transactions on Image Processing 27.3 (2017): 1323-1335.
19. Freund, Y., Schapire, R.E.: A short introduction to boosting. In: In Proceedings of the Sixteenth IJCAI. pp. 1401–1406. Morgan Kaufmann (1999)
20. Zhao, Hong, and Shenglong Yu. "Cost-sensitive feature selection via the $l_{2,1}$-norm." International Journal of Approximate Reasoning 104 (2019): 25-37.
21. He, He, Hal Daumé III, and Jason Eisner. "Cost-sensitive dynamic feature selection." ICML Inferning Workshop. 2012.
22. Metzler, Guillaume, et al. "Tree-based cost sensitive methods for fraud detection in imbalanced data." Advances in Intelligent Data Analysis XVII: 17th International Symposium, IDA 2018,'s-Hertogenbosch, The Netherlands, October 24–26, 2018, Proceedings 17. Springer International Publishing, 2018.
23. Pudil, P., Novovičová, J., and Kittler, J. (1994). Floating search methods in feature selection. Pattern recognition letters, 15(11) :1119–1125.
24. Kaur, S. and Cheema, S. S. (2017). Big data and analysis of weather forecasting system. International Journal of Advanced Research in Computer Science, 8(7).
25. Alberto Fernández, Salvador García, Mikel Galar, Ronaldo C Prati, Bartosz Krawczyk, and Francisco Herrera. Learning from imbalanced data sets. Springer, 2018.
26. Bhapkar, Yogita. (2022). Evaluation of K-NN and Decision Tree Classifiers for Classification of Home Loan Customers Using Data Mining Classification Technique.
27. Tang, J., Alelyani, S., and Liu, H. (2014). Feature selection for classification : A review. Data classification : Algorithms and applications, page 37.
28. Neetha, K. S., & Narayan, D. L. (2024). Feature selection using adaptive manta ray foraging optimization for brain tumor classification. Pattern Analysis and Applications, 27(2), 29.