Universidade Estadual de Campinas
Instituto de Computação

Athyrson Machado Ribeiro

CI-EX: Confident-Inline Extrapolation for Rejection Inference in Financial Credit Scoring

CI-EX: Confident-Inline Extrapolation para Inferência de Rejeitados em Pontuação de Crédito Financeiro

CAMPINAS
2024

Athyrson Machado Ribeiro

CI-EX: Confident-Inline Extrapolation for Rejection Inference in Financial Credit Scoring

CI-EX: Confident-Inline Extrapolation para Inferência de Rejeitados em Pontuação de Crédito Financeiro

Dissertação apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Dissertation presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Master in Computer Science.

Supervisor/Orientador: Prof. Dr. Marcos Medeiros Raimundo

Este exemplar corresponde à versão final da Dissertação defendida por Athyrson Machado Ribeiro e orientada pelo Prof. Dr. Marcos Medeiros Raimundo.

CAMPINAS
2024

Ficha catalográfica
Universidade Estadual de Campinas
Biblioteca do Instituto de Matemática, Estatística e Computação Científica
Ana Regina Machado - CRB 8/5467

R354c

Ribeiro, Athyrson Machado, 1997-
CI-EX : Confident-Inline Extrapolation for rejection inference in financial credit scoring / Athyrson Machado Ribeiro. – Campinas, SP : [s.n.], 2024.

Orientador: Marcos Medeiros Raimundo.
Dissertação (mestrado) – Universidade Estadual de Campinas, Instituto de Computação.

1. Aprendizado de máquina. 2. Sistemas de credit scoring. 3. Inferência de rejeitados. I. Raimundo, Marcos Medeiros, 1988-. II. Universidade Estadual de Campinas. Instituto de Computação. III. Título.

Informações Complementares

Título em outro idioma: CI-EX : Confident-Inline Extrapolation para inferência de rejeitados em pontuação de crédito financeiro
Palavras-chave em inglês:
Machine learning
Credit scoring systems
Reject inference
Área de concentração: Ciência da Computação
Titulação: Mestre em Ciência da Computação
Banca examinadora:
Marcos Medeiros Raimundo [Orientador]
Sandra Eliza Fontes de Avila
Eliezer de Souza da Silva
Data de defesa: 24-04-2024
Programa de Pós-Graduação: Ciência da Computação

Identificação e informações acadêmicas do(a) aluno(a)
- ORCID do autor: https://orcid.org/0009-0001-4494-4607
- Currículo Lattes do autor: https://lattes.cnpq.br/8910332182361384

Universidade Estadual de Campinas
Instituto de Computação

Athyrson Machado Ribeiro

CI-EX: Confident-Inline Extrapolation for Rejection Inference in Financial Credit Scoring

CI-EX: Confident-Inline Extrapolation para Inferência de Rejeitados em Pontuação de Crédito Financeiro

Banca Examinadora:

- Prof. Dr. Marcos Medeiros Raimundo
  Universidade Estadual de Campinas
- Profa. Dra. Sandra Eliza Fontes de Avila
  Universidade Estadual de Campinas
- Dr. Eliezer de Souza da Silva
  Fundação Getulio Vargas

A ata da defesa, assinada pelos membros da Comissão Examinadora, consta no SIGA/Sistema de Fluxo de Dissertação/Tese e na Secretaria do Programa da Unidade.

Campinas, 24 de abril de 2024

All progress is based upon
a universal innate desire
of every organism
to live beyond its income.

(Samuel Butler)

Agradecimentos

Em primeiro lugar gostaria de agradecer à minha família por todo o apoio que recebi ao longo da minha vida. Agradeço à minha mãe, Carmen Lucia, e ao meu pai, Antônio Gomes, por terem sempre me elevado, com muito amor, aos lugares necessários à minha formação acadêmica. Ao meu irmão, João Paulo, por todo companheirismo e inspiração. Àos meus avós e tios, em especial à minha avó Ana Rosa, por ter sido sempre o maior exemplo de amor em minha vida.

Gostaria de agradecer ao meu orientador Marcos M. Raimundo, por todos os ensinamentos, oportunidades, apoio e amizade.

Sou imensamente grato à Unicamp, especialmente aos Laboratórios Recod.ai e Hiaac, por toda estrutura física e redes de conhecimento fornecidas, e especialmente por todos os amigos que pude encontrar nesses valiosos espaços. Gostaria de destacar especialmente, mas não exclusivamente: Aline, Arthur Salles, Arthur Hendricks, Augusto Cézar, Beatriz, Caio, Daniel, Giorgio, Giovanni, Jamila, Jansem, João, Juan, Leo, Levi, Luã, Matheus, Rafael Werneck, Thalita, e Wladimir.

Finalmente, gostaria de expressar meus agradecimentos ao meu amigo Chico, por sempre acreditar em meu potencial e me incentivar a me inscrever no programa de mestrado da Unicamp.

Este projeto foi apoiado pelo Ministério da Ciência, Tecnologia e Inovações do Brasil, com recursos concedidos pela Lei Federal 8.248, de 23 de outubro de 1991, sob o PPI-Softex. O projeto foi coordenado pela Softex e publicado como Agentes inteligentes para plataformas móveis baseados em tecnologia de arquitetura cognitiva [01245.013778/2020-21]. Este estudo também foi financiado, em parte, pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

Resumo

Um dos principais desafios no campo de pontuação de crédito é a indisponibilidade de informação sobre a capacidade de pagamento de clientes que tiveram suas solicitações de crédito negadas (clientes rejeitados). A maioria dos modelos básicos de pontuação de crédito considera apenas a população de clientes aceitos, o que pode introduzir viés contra indivíduos fora dessa distribuição. Para enfrentar esse viés, métodos de Inferência de Rejeitados (RI) visam inferir informações faltantes de indivíduos rejeitados e assim integrá-los ao sistema de pontuação de crédito. Métodos tradicionais de inferência de rejeitados na literatura frequentemente presumem a viabilidade da estratégia de extrapolar o comportamento de clientes rejeitados a partir de dados de clientes aceitos. Apesar das possíveis diferenças nas distribuições de dados entre esses grupos. Portanto, para mitigar a extrapolação cega entre clientes aceitos e rejeitados, introduzimos um novo framework de Confident Inlier Extrapolation framework (CI-EX). Primeiramente, o CI-EX identifica distribuições das amostras dos rejeitados de forma iterativa usando um modelo de detecção de outliers. Em seguida, atribui rótulos aos indivíduos rejeitados mais próximos da distribuição da população aceita, com base em probabilidades derivadas de um modelo supervisionado de classificação. Especificamente, apenas as amostras para as quais o modelo fornece maior confiança na previsão são incorporadas ao novo conjunto de dados de treinamento, abordando assim os viéses de extrapolação no processo de inferência. Além disso, propomos o framework Confident Inlier Label Spreading (CI-LS), onde rótulos para amostras rejeitadas são inferidos usando um modelo de classificação semi-supervisionado. A eficácia de nossos métodos propostos é validada através de experimentos realizados nos conjuntos de dados de crédito HomeCredit e Lending Club. Os resultados são avaliados usando a Área Sob a Curva (AUC), uma métrica muito relevante em crédito, e métricas específicas de RI como Kickout e a métrica introduzida neste trabalho, denominada Área Sob o Kickout (AUK). É importante notar que a avaliação da AUC é baseada exclusivamente em amostras de clientes aceitos. Nossos resultados demonstram que os métodos de RI, incluindo os frameworks propostos, geralmente envolvem um trade-off entre as métricas AUC e RI. No entanto, nossos métodos consistentemente superam os modelos de RI existentes na literatura de crédito em termos de métricas específicas de RI na maioria dos experimentos.

Abstract

One of the main challenges in the field of credit scoring is the unavailability of the repayment capacity data of clients who have had their credit applications denied (rejected clients). Most basic credit scoring models only consider the population of accepted clients, potentially introducing bias against individuals outside of that distribution. To address this bias, Reject Inference (RI) methods aim to infer missing information from rejected individuals and integrate them into the credit scoring system. Traditional reject inference methods from the literature often assume the feasibility of extrapolating the behavior of rejected clients from accepted client data, despite potential differences in data distributions between these groups. Therefore, to mitigate blind extrapolation between accepted and rejected clients, we introduce a novel Confident Inlier Extrapolation framework (CI-EX). Initially, CI-EX iteratively identifies the distributions of samples from rejected clients using an outlier detection model. It then assigns labels to rejected individuals closest to the distribution of the accepted population based on probabilities derived from a supervised classification model. Specifically, only samples for which the model gives higher prediction confidence are incorporated into the new training dataset, thus addressing extrapolation biases in the inference process. Additionally, we propose the Confident Inlier Label Spreading framework (CI-LS), where labels for rejected samples are inferred using a semi-supervised classification model. The effectiveness of our proposed methods is validated through experiments conducted on the HomeCredit and Lending Club credit datasets. Results are evaluated using the Area Under the Curve (AUC), a pertinent metric in credit, and RI-specific metrics such as Kickout and the novel metric introduced in this work, denoted Area under the Kickout (AUK). It is important to note that AUC evaluation is based exclusively on accepted client samples. Our findings demonstrate that RI methods, including the proposed frameworks, generally involve a trade-off between AUC and RI metrics. However, our methods consistently outperform existing RI models from the credit literature in terms of RI-specific metrics across the majority of experiments.

List of Figures

List of Tables

Lista de Siglas e Conceitos

A-DW	Downward Augmentation Model
A-FU	Fuzzy-Augmentation Model
A-SC	Augmentation with Soft Cut-Off Model
A-UW	Upward Augmentation Model
Accepts	Accepted clients data
AUC	Area Under the Curve
AUK	Are Under the Kickout
BM	Benchmark model: Model trained with only data from accepted set.
CI-EX	Confident-Inline Extrapolation for Rejection Inference
CI-LS	Confident-Inline Label Spreading for Rejection Inference
E-C	Confident Extrapolation Model
Inlier	Sample that belongs to a dataset distribution
LSP	Label Spreading Model
OD	Outlier Detection
Outlier	Sample that differs too much from the samples of a dataset distribution
PAR	Parcelling Model
Rejects	Rejected clients data
RI	Reject Inference

Contents

Chapter 1

Introduction

1.1 Motivation

Credit Scoring affects the vast majority of people worldwide. Whenever a client requires a loan (or a credit card), the bank calculates their credit score to evaluate their ability to pay their debts (Mester et al., 1997). The fear of granting loans to many *default* applicants leads companies to use harsh credit scoring policies. Such lending standards could exclude many good debtors and exacerbate harm to underrepresented communities. A credit model trained with only a small and under-representative subset of society is not ideal to classify the whole population reasonably and precisely. This phenomenon is known as sample bias and it occurs when a credit model is trained with only accepted clients (Guo et al., 2023; Nikita Kozodoi et al., 2019; Kang et al., 2021). Due to harsh policies applied by the companies, most applicants are denied a loan, which means, in many cases, the rejected applicants constitute the majority of data in credit datasets(Kang et al., 2021; Shen et al., 2020). There are some approaches which assimilate information from both rejected and accepted clients to improve credit scoring systems. This group of techniques is called Reject Inference (RI). The incorporation of RI techniques grants substantial advantages: (1) A considerable decrease of sample bias — coming from more robust models of credit scoring trained with information of a bigger population; (2) Minimization of data waste; (3) Better evaluation of marginalized communities.

RI literature has advanced considerably in the last few decades and many papers were published highlighting the importance of RI application on the credit scoring process. From simple assumptions, considering all reject as bad cases (potential defaults) to an entire network using rejected clients' information to classify credit scoring (Siddiqi, 2017; Liu et al., 2022b). However, there are some strong assumptions in RI literature that can not be ignored. The first one is that the behavior of the rejected population can be extrapolated based on the accepted population. This is often not the case, as there are many differences in the distribution of accepted and rejected clients. The second assumption is that a small gain in accuracy is the objective of RI applications (Sabato, n.d.). When the entire pipeline, from training to testing, is based solely on the accepted population, credit scoring models can already have high predictive accuracy. However, we believe ignoring the existence of sample bias is not a good way to tackle credit scoring, as many people historically outside the distribution of the accepted population can be

harmed.

A large number of recent papers in RI literature propose frameworks combining several RI and machine learning techniques to label and filter out samples. This combination seems to lead to models with high classification power (Shih et al., 2022; Shen et al., 2020; Liao et al., 2022). However, most of the RI literature is based on at least one of the two assumptions mentioned before. This research proposes two novel frameworks consisting of several verification steps to assure confidence in the RI process utilize. Confident-Inline Extrapolation for Rejection Inference (CI-EX) uses outlier detection and classification probabilities to label and filter the most confident samples, and Confident-Inline Label Spreading for Rejection Inference (CI-LS) is similar but applies a semi-supervised technique called Label Spreading (Zhou et al., 2003) instead of a classifier. They are built on an iterative procedure, where each iteration implies a new model more inclusive of the RI population than its predecessor. This is made to avoid the extrapolation bias. We tackle the second assumption problem by using metrics that take into consideration the RI population. We argue that these metrics are more suited to evaluate the actual performance of RI techniques. We evaluate our methods with the Reject Inference metric kickout, presented by Kozodoi et al. (2020) as having a higher correlation with correctly evaluating the unbiased population. We also propose a new metric for RI based on kickout, called Area under Kickout (AUK). Our proposed techniques consistently outperform other Reject Inference techniques in the literature at these RI metrics.

1.2 Hypotheses

This work is motivated by the following hypotheses:

- Accepted clients data is not enough to fairly train a credit scoring model;
- It is possible to infer the behaviour of rejected clients data using the data from the accepted clients;
- Evaluating RI models with only data from accepted clients can not effectively reflect the RI models true performance;
- Using data from rejected clients it is possible to create a credit scoring model than performs better than a model that is trained with only accepted clients data.

1.3 Contributions

To address our hypotheses, we propose two frameworks based on Reject Inference for credit scoring CI-EX and CI-LS and evaluate these framework with metrics that take into consideration data from both accepted and rejected clients data. More specifically, our contributions are:

- The semi-supervised framework Confident-Inline Extrapolation for Rejection Inference (CI-EX);

- A variation of our proposed semi-supervised framework CI-EX, called Confident-Inline Label Spreading for Rejection Inference (CI-LS);
- The proposition of Area Under the Kickout metric (AUK) for Reject Inference models;
- The evaluation of classical Reject Inference models from literature with metrics that take into consideration both accepted and rejected clients.

1.4 Outline

This dissertation is organized as follows. Chapter 2 presents an introduction to Reject Inference concepts and literature, as well as other related concepts to this research like credit scoring and outlier detection. Chapter 3 presents the motivation and details of the structure of our proposed framework. Chapter 4 presents the detailing of our experiments structure, the datasets used on this research, as well as the evaluation metrics considered. Chapter 5 presents the results of our experiments as well as our considerations about them. And finally Chapter 6 presents our conclusions about this research, limitations found, and plans for future works.

Chapter 2

Background

2.1 Credit Scoring

Credit scoring is critical to many processes in granting loans, leasing properties, and other commodities. The decision to approve or to deny a loan to a borrower hinges on their ability to convincingly assure the lender of their trustworthiness (Anderson, 2022). However, if this decision is made without a protocol or transparency, many problems can arise. The most obvious problem is the financial loss caused by lending funds to borrowers who will not repay them. They are traditionally called bad payers in credit scoring literature (the borrowers who pay back on time are called good payers). Therefore, implementing an automatic, or at least semi-automatic, trustworthiness system is crucial. This system is known as credit scoring (Kang et al., 2021). For simplicity, without loss of generality, from now on, we will limit our discussion to the process of credit scoring that involves a company lending funds to an individual.

The credit scoring process generally involves obtaining information about an individual and comparing this information to other individuals, from which we have payment behavior data. In machine learning, this information about an individual is called features, and the classification of whether the individual is a good or bad payer is called class or target. The assumption is the payment behavior of an individual can be estimated based on their features. The features that may assist in this estimation are often related to the client's economic situation, the loan itself, or the individual's historical credit data (which, in many cases, is unavailable to the company). With the use of these features and respective targets, classification models can be fitted by the company to assist in the process of selecting trustworthy clients to grant loans.

2.2 Reject Inference

When building a classifier to automate the decision of who should be worthy of receiving a loan, an essential requirement is that such a classifier is good at generalization. In realistic terms, such a model should perform well even with data that differs, to some extent, from the data it was trained upon. This is why we separate the data between training, validation, and testing when training Machine Learning models. The model's

generalization directly relates to how much the data it was fitted reflects the real world in which it will be applied. Therefore, when only data from accepted clients is used in training the credit pipeline, as illustrated on Figure 2.1 a), a clear sample bias is identified. Figure 2.1 a) illustrates a credit pipeline from a company that builds its classification model based only on approved clients from previous iterations. However, not only approved clients, but also the population rejected by earlier iterations as well as clients coming from unseen distributions may ask for a loan from this company. Therefore, we have a model based on a sample that does not accurately reflect the entire population, resulting in what is known as sample bias. At each iteration of this pipeline, the sample bias will only grow, leading the company to use models of classification that are less applicable to the entire population each time (Siddiqi, 2017).

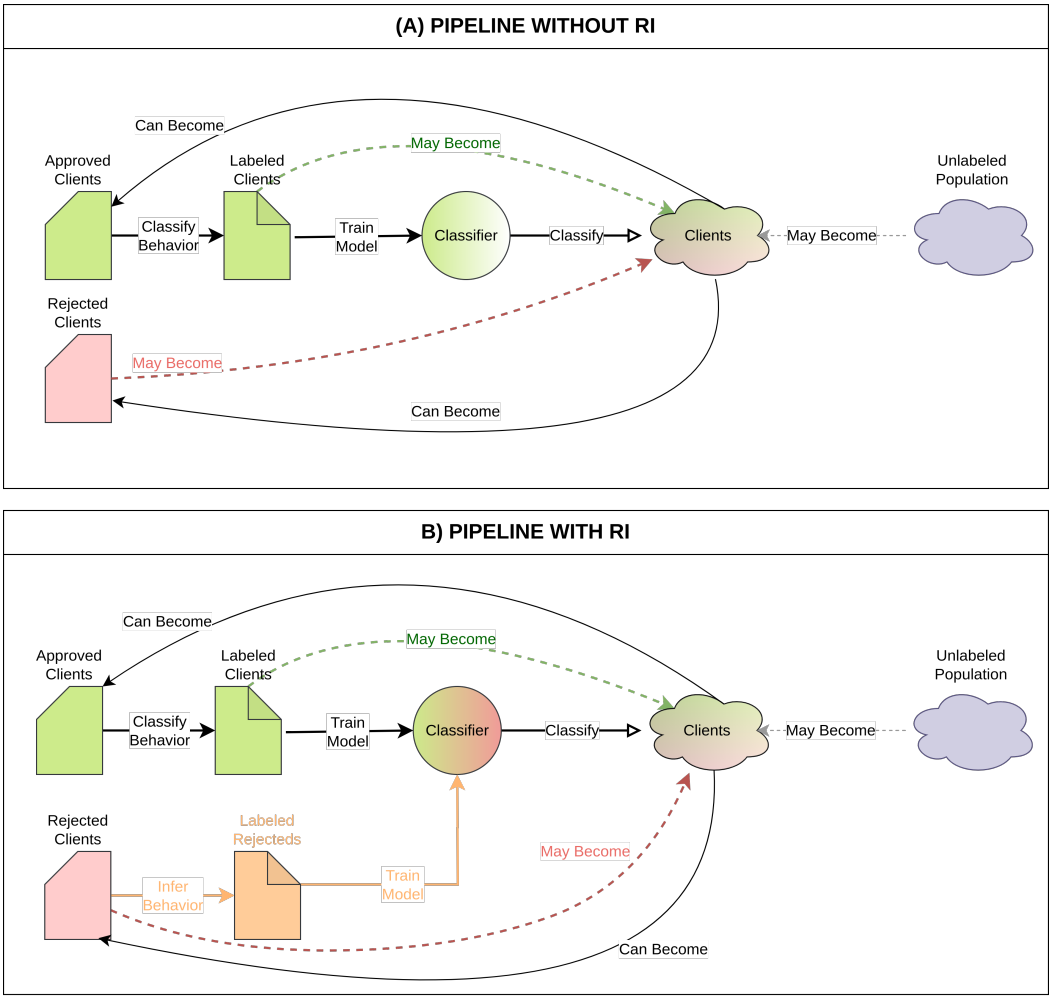

Figure 2.1: (A) Pipeline that discards rejected clients data, versus (B) pipeline that applies Reject Inference.

The biggest obstacle in avoiding sampling bias in credit scoring is the lack of labels for the rejected clients. Approved clients, as illustrated in Figure 2.1, can have their behavior observed. Depending on whether they repay the loan within the stipulated time, for example, they can be almost accurately classified as good or bad payers. The

same cannot be applied to rejected clients. The company has their data when they asked the loan, but can not retrieve accurate labels based on their repayment behavior. One solution would be to approve all clients and classify them according to their behavior (Anderson, 2022). However, this would be too costly for most loan companies. Luckily, there are more applicable solutions from both literature and business. These solutions are known as Reject Inference (RI), and most of them can be described as making the classification model aware of the rejected client population. Figure 2.1 b) illustrates RI in a credit pipeline. As shown in the Figure 2.1 b), the behavior of the rejected clients is inferred through some technique, and a label is given to them based on that. The data from these clients is then concatenated to the training set to build the new classifier. The model resulting from this process is a model which has more knowledge of the whole population in comparison to the model built with only accepted clients.

However, Reject Inference is not without flaws and caveats. First, it should be mentioned that there are other approaches to the technique, other than using it to inflate the training set, some of which will be described in the next subsections. Second, RI and statistical processes are built on a series of assumptions, such as the type of missing data problem, the viability of inferring the missing features and labels of rejected clients, and the evaluation process capable of measuring the actual performance of the credit pipelines.

Reject Inference (RI) techniques vary significantly in incorporating the rejected data into the credit model pipeline. Even RI techniques in the same family can have very different approaches, as is the case for augmentation techniques (Siddiqi, 2017). Because of such diversity, there is no consensus on the best technique for all scenarios. Each technique also has flaws and restraints (Crook and Banasik, 2004). Despite their limitations, the application of an RI technique should bring a credit scoring model that is more robust, less biased toward the whole population, and less wasteful of data.

Many authors (El Annas et al., 2022; Shen et al., 2020; Liu et al., 2022b; Anderson, 2022) mention the three types of missingness of data proposed by Little and Rubin (2019) when introducing the lack of rejected data in most credit scoring systems (El Annas et al., 2022):

- Data can be missing due to completely random reasons (MCAR) when there is not any relation between the missingness of the data and any other variable related to the system or sample;
- Data can be missing at random (MAR) when there is a relation between the missingness of the variable of interest and some other variable in the dataset which is not the variable of interest;
- Data can also be missing not at random (MNAR), when the missingness is related with the missing data itself and may be caused by some unobserved variables.

According to Liu et al. (2022b), MNAR can play a significant role in RI due to the subjective reasons that can influence the approval of a loan in not fully automated credit scoring systems. Anderson (2022) also affirms that most cases of missing data in credit scoring systems can be attributed to MNAR due to the outside factors that can not

be represented in a credit model but influence the decision of which applicants will be rejected.

The RI technique can be applied in different stages of the model pipeline. Maybe the most intuitive approach would be to infer the labels of the rejected clients to eventually expand the training set with their data, like extrapolation (Siddiqi, 2017), parceling (Siddiqi, 2017) and label spreading (Zhou et al., 2003): the Data Inflating Methods. Some techniques, however, only apply the rejected data in the form of adjusting the weights of the credit model, which is the case for most types of augmentation (Siddiqi, 2017; Anderson, 2022): the Weight Adjusting Methods. Some authors go a step further and propose new machine learning models built to consider the existence of rejected data (Liu et al., 2022b): the Model Approach Methods.

Siddiqi (2017) explains that the usefulness of RI techniques is highly linked to our confidence in our previous system for Approval/Rejection of loans. RI may not be indicated if the confidence is too low, as in close to decided randomly, or too high, together with a high approval rate. Although it is not a recommended strategy, if the confidence is too high, one straightforward RI technique that can be applied is to assume all rejects as bad payers (Siddiqi, 2017; Anderson, 2022). There are, however, many reasons for the application of RI techniques. The most known reason is to avoid sample bias by using a subset not truly representative of the whole population (Siddiqi, 2017; Kang et al., 2021; Song et al., 2022; Shen et al., 2020). Another strong reason for applying RI techniques is to fix past decisions made in credit scorecard development. For example, RI can help make marginalized individuals more considered and less prejudiced in the credit process (Siddiqi, 2017). For financial institutions, RI can inform a more accurate default rate of the population, avoiding monetary losses (Liao et al., 2021; Siddiqi, 2017). The following subsections describe the three groups of RI techniques mentioned in this section.

2.2.1 Weight Adjusting Methods

Augmentation, also known as Reweighing, is a technique where the weights of the accepted data are adjusted to consider the probabilities of rejection (Siddiqi, 2017; Anderson, 2022). An Approval/Rejection (AR) model is fitted with accept and reject status being used as the classes. The model is then applied to the accepted data, and the probability of each accepted sample is retrieved. In Upward Augmentation (A-UW), the new weight is calculated by equation 2.1, while in Downward Augmentation (A-DW), the new weight is calculated by equation 2.2. Where $\hat{w}$ is a new weight, w is the previous weight (we can assume one as its value), and $p(A)$ is the probability of being accepted given by the AR model (Anderson, 2022).

$$\hat{w} = \frac{w}{p(A)} \tag{2.1}$$

$$\hat{w} = w \cdot (1 - p(A)) \tag{2.2}$$

Another way to use Augmentation is to sort the accepted and rejected samples by the $p(A)$, then separate these samples in n splits according to the $p(A)$. For each split, the proportion of accepts between accepts and rejects contained in that split is calculated

($AF = \frac{nA}{nA+nR}$). Then the augmentation factor for that split will be $\frac{1}{AF}$. The AF will then be used as the new weight for all accepted samples in that split. This technique is called Augmentation with Soft Cut-Off (A-SC) (Siddiqi, 2017; Ehrhardt et al., 2020). One more very well-known Augmentation method is Fuzzy-Augmentation (A-FU), also known as Fuzzy-Parcelling (Anderson, 2022). A differential of this technique is that it is both a Data Inflating Method and a Weight Adjusting Method. In this technique an AR model is also fitted, however, the rejected data is concatenate to the new dataset twice. First, it is appended receiving 0 as a label and $p(A)$ as weight, then it is again appended but with 1 as a label and $p(R)$ (probability of rejection) as weight. The accepted samples receive 1 as weight.

2.2.2 Data Inflating Methods

The use of information about the labeled (accepted) data to infer the labels of the non-labeled (rejected) data is known as Extrapolation (Anderson, 2022). Simple extrapolation techniques use a classifier fitted in the accepted data to infer the labels of the rejected data. Suppose we assume our classifier is good enough to do this inference process. In that case, we can use the inferred labels for the rejected samples as actual labels and concatenate the rejected samples in the new training set. However, it may not be wise to append all the rejected samples at once to the new training set. If we are interested in balance the number of bad payers in the training set we could, for example, add only the samples inferred as bad payers from the rejected group, we will call this alternative "Bad Extrapolation" (BE). Another choice would be to consider our confidence in the predictions of our extrapolation model, and to add only the samples most far from the classification threshold, we will call this alternative "Confident Extrapolation" (E-C).

Instead of using a fitted classifier to infer the labels of the rejected data, we can infer the labels of the rejected data alongside the training of a label-spreading classifier. Proposed by Zhou et al. (2003), this technique relies upon the assumption that nearby samples in a dataset are inclined to have the same labels. After the label spreading classifier is fitted, we can retrieve the labels attributed to the rejected samples by the model. Then, we can expand the training set by concatenating the rejected samples labeled by the label spreading classifier to the training data. We will abbreviate this technique as LSP.

A technique similar to ASC is Parcelling (PAR) (Siddiqi, 2017). However, instead of changing the weights of accepts, we use the splits to label the rejected samples in this technique. First, a classifier is fitted with accepted data. This classifier is then used to calculate the probability of default on both accepts and rejects. These samples are then sorted based on their probability of default and split based on score intervals. The number n of score intervals is an arbitrary parameter. For each split, we calculate the ratio of true bad payers (β) between all accepts included in that split. But, since we are interested in labeling the rejects, we multiply the bad rate by a prejudice factor ρ. With the updated bad rate ($\hat{\beta}$), we can calculate the new expected good rate: $\hat{\kappa} = 1 - \hat{\beta}$. The rejected samples in the split are then randomly assigned a label in proportion to the updated good and bad rate for that split. Once this process is concluded for all splits, the rejected samples can be concatenated to the new training set.

2.2.3 Model Approach Methods

A more recent approach to RI is the creation of machine learning models built specifically to work with both accepts and rejects. In their work, Liu et al. (2022b) propose a Reject Aware Multi-Task Network (RMT-Net) that takes into consideration the high correlation between the tasks of classification between approval/rejection and default/non-default clients to improve its learning capabilities. Another RI network, proposed by Guo et al. (2023), Transductive Semi-Supervised Metric Network (TSSMN) consists of the union of two networks, the first one is responsible to map the samples into a metric space. The second one uses transductive label propagation to label the samples according to the proximity given by the first network.

2.3 Outlier Detection

Outlier Detection is a relevant concept in machine learning. An outlier is a sample that differs too much from the samples of a distribution, which implies it does not belong to that distribution. Subsequently, an inlier is seen as a sample that belongs to that distribution on which the outlier detection (OD) algorithm was trained (Xia, 2019; Ali et al., 2023). Generally, removing outliers from the training dataset is expected to translate to a model's higher performance. Therefore, the OD models, such as Isolation Forest (Liu et al., 2008), are usually employed to identify outliers samples that should be removed from the dataset. However, some authors have found OD as a tool for more ambitious tasks (Xia, 2019; Nikita Kozodoi et al., 2019; Coenen et al., 2020).

Since data for rejected clients does not contain ground truth labels, OD algorithms are well suited for RI techniques because most are based on unsupervised learning, which does not require labels for training (Xia, 2019). In their work, Xia (2019) proposed using OD as a Data Inflating Method for RI. They use Isolation Forest to label samples in the rejected dataset. Outliers in the rejected dataset are seen as samples that should not have been rejected and are reclassified as suitable applicants. The inliers are seen as samples that are correctly rejected and should be classified as bad applicants. The authors claim to be the first to employ OD as a RI technique, and their work inspired others.

Another combination of OD and RI techniques is found in the works of Nikita Kozodoi et al. (2019), Coenen et al. (2020) and more recently in Shih et al. (2022). Nikita Kozodoi et al. used OD to iteratively identify inadequate samples from the rejected dataset based on the distribution of the accepted population, ignoring those samples too close and too far from the accepted population. Where Coenen et al. approach was to use OD to reclassify samples from both the accepted and rejected population in the pre-processing stage. Accepted samples marked as outliers were removed from the accepted dataset, and outliers in the rejected population were incorporated into the training set as suitable applicants. Shih et al. followed an approach more similar to Xia, however. In their work, OD was applied to identify potential good cases between the rejected population and remove potential bad cases from the accepted population, effectively using OD for relabeling samples.

2.4 Related Work

In their work, Xia (2019) proposed one of the first applications of outlier detection in RI. Different from most works at the time, they applied outlier detection after the preprocessing phase of the pipeline to label rejected samples. However, although the authors criticized previous literature assumptions on direct extrapolation of behaviours from the accepted to rejected population, they also applied outlier detection with a similar principle. Labeling all outliers of the rejected group as good payers, they assumed the entire rejected population can be directly divided between good and bad payers. However, according to Coenen et al. (2020), not all samples from the rejected group can be reliable labeled based i.e. there will be some cases where there will just be not enough information to infer the label of a sample based on its features. Besides, the number of individuals selected as outliers in the reject set will be influenced by the contamination threshold predefined, and if this is the only criteria utilized the number of inferred good payers between the rejected population can be vastly exaggerated. Despite that, their work achieved great results, surpassing the models trained with only accepted samples, and influenced others in the RI literature (Coenen et al., 2020; Shih et al., 2022).

More recently, Shih et al. (2022) proposed a similar application of outlier detection for RI. Adding to the method proposed by Xia (2019), the authors ruled that outliers among rejected samples would be classified as good payers, whereas outliers in accepted samples should actually be excluded from the training set (as rejects). Another contribution from the authors was the use of K-nearest neighbor to fill in missing features in the rejected dataset, addressing the significant discrepancy between the number of features in the accepted and rejected datasets. From this combination of techniques the authors achieved great results with their proposed framework at the Lending Club dataset. However, the authors only measured the performance of their techniques solely on accepted samples and utilized of features that could only be obtained after the loan approval phase.

In their work, Liao et al. (2021) applied a self-training method for RI where rejected samples would be iteratively added to the training set and labeled based on its prediction confidence. The authors claim that labeling data with low certainty has low chance of improving classification performance. With this, the authors were able to augment the training dataset by 126% and achieve better results in the majority of experiments than the model trained with approved only samples and others RI methods studied. However, the authors demonstrated their results only with one private and relatively small sized dataset.

Outlier detection, as an unsupervised method, has a great potential in reject inference, where labels of most of the data are missing. However, we identified a gap in the RI literature, as no work has yet applied outlier detection in an iterative and controlled manner. This could help avoid biased assumptions that affect most extrapolation methods, which, despite their flaws, remain one of the most promising groups of RI techniques.

Chapter 3

Proposed Framework

In this research, we propose a novel framework for RI that presents a semi-supervised learning method combining outlier detection (OD) and a confidence rule to infer the unlabeled sample classes. We call this framework Confident-Inline Extrapolation for Rejection Inference (CI-EX). Our approach, as many other studies involving RI and OD, is inspired by the methodologies of Xia (2019). However, we do not use OD as a classification tool for the rejected data. Instead, we chose an approach similar to that of Nikita Kozodoi et al. (2019), who also proposed an iterative method. However, unlike their approach, we did not use OD to filter out outliers but to select inlier samples at each iteration. And, differently from Shih et al., we propose an iterative method in which OD is not the actual labeler tool but only a filter step.

Table 3.1: Mathematical notations for Algorithm 1 and 2

Notation	Description
X_{train}	set with labeled data
Y_{train}	set with labels
X_{rej}	set with unlabeled data
η	the number of samples to be added
ρ	ratio expected between good and bad payers
c	desired number of samples to be retrieved
Δ	class (0 - non-default, 1 - default)
X_{Θ}	set with inliner samples
X_{Δ}	set with retrieved data
Y_{Δ}	inferred labels for retrieved data
$X_{train_{\Delta}}$	set with data labeled as Δ
x_j	feature vector of example j
$P(X_{rej} = \Delta)$	probability of X_{rej} being Δ

To identify the samples from the rejected set we are most confident are from a specific class Δ, we propose an algorithm that performs a two-step verification on each sample. First, we check if the rejected sample is a non-outlier for the class Δ. Then we check if that sample belongs to the subset of c samples with the highest probability of belonging to that class. At each iteration, our algorithm labels and adds η samples from the unlabeled

set to the training set and removes those samples from the unlabeled set.

3.1 Retrieve Confident Samples

The Retrieve Confident Samples Algorithm (Algorithm 1) describes the core of our framework, and Table 3.1 constitutes of a quick guide for better reading of our proposed algorithms. The algorithm takes as input a labeled training dataset, where X_{train} represents the informative features of the dataset, and Y_{train} represents the target feature of the dataset. Due to changes during method iterations, Y_{train} may consist of both ground truth labels and inferred labels, and X_{train} may consist of both accepted and rejected client data. The proportion between accepted and rejected data will depend on the current iteration of the framework as new data is added to the training set.

As mentioned before, our framework employs a two-step verification to ensure the rejected samples added at each iteration have a bigger probability of being the ones we are most confident should get the inferred labels. The first step uses Isolation Forest (Liu et al., 2008), an outlier detection algorithm, to divide the rejected samples between outliers and non-outliers. Instead of fitting the model with the entire training set, we fit the model with one class at each time from the training set (X_{train_Δ}). Our first hypothesis is that samples considered non-outliers based on X_{train_Δ} are likelier to belong to that class. In Algorithm 1, this set of samples considered non-outliers, X_Θ, move on to the next step. We have experimented with two modes for labeling the rejected set, which will be described subsequently.

Extrapolation Mode

We call this version of the proposed framework Confident-Inline Extrapolation (CI-EX). Figure 3.1, illustrate how this version of Algorithm 1 works. As can be seen in the figure, and in the step 2 and 3(a) of the algorithm, in the CI-EX mode, the current training data is used to train the Isolation Forest algorithm and the classifier. However, although the whole training data is used to train the classifier, only the samples from the respective class Δ are used to train the Isolation Forest. Because of this, the Algorithm 1 needs to be executed twice at each iteration returning c samples with label Δ — or less if fewer than the stipulated number of samples match the criteria. In step 4, using the Isolation Forest, we classify the Rejects Data into outliers and inliers. Outiers are ignored temporally, but will belong to the updated rejects dataset at the end of the iteration of the Algorithm 1. Inliers, however, are further subdivided into top confident, which go to set X_Θ, and less confident samples.

With this strategy, the step 5.1 of Algorithm 1 uses the probabilities derived from a classifier with balanced weights[1] to label the inliers samples and filter the c most confident samples[2] (steps 5.2 to 5.3). However, the less confident samples, will also become part of

[1] In our implementation, instead of using the default learning procedure that makes all samples equally important, the weight of each sample is inversely proportional to the number of samples of its class.

[2] Since our classifier uses balanced weights, we can use 0.5 as the threshold to classify the samples between *good* and *bad* cases.

the updated rejects dataset at step 5.4 of the iteration of the Algorithm 1. The algorithm then returns c samples with label Δ — or less if fewer than the stipulated number of samples match the criteria. After the Algorithm 1 is executed for both classes, the rejected dataset and the training dataset are updated, as illustrated in Figure 3.1.

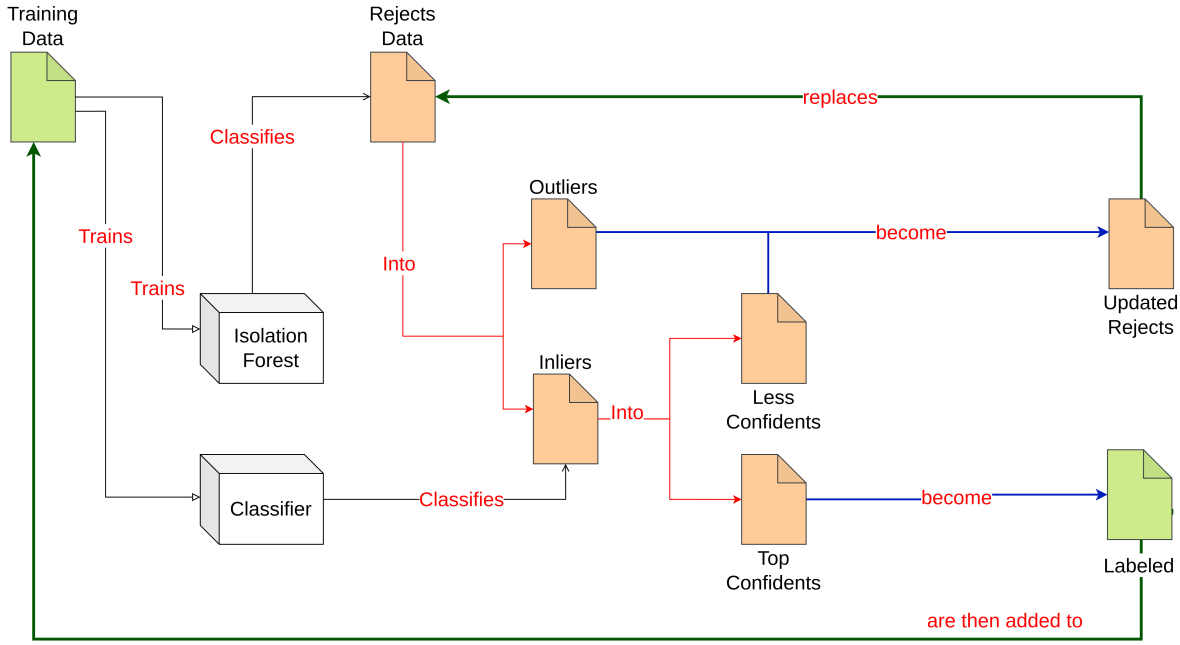

Figure 3.1: Representation of CI-EX framework to perform Reject Inference.

Label Spreading Mode

We call this version of the proposed framework Confident-Inline Label Spreading (CI-LS). Figure 3.2, illustrate how this version of Algorithm 1 works. As the figure shows, the algorithm's execution proceeds in a very similar way to the CI-EX mode. However, instead of using a typical classifier to label our samples after the fitting process, we get the labels directly from the training process with this strategy, using a semi-supervised model called Label Spreading (Zhou et al., 2003). In step 3(b) the label spreading algorithm is fitted with both the labeled and unlabeled samples. The unlabeled samples receive -1 as a label and, during the fitting process, receive a score from the `label_distribution_` function. Similarly to the extrapolation mode, in step 5.2 of the algorithm, we use this score to choose the most confident samples. The following steps of the algorithm are precisely as described for the extrapolation mode.

3.2 Expand Dataset

The Expand Dataset Algorithm (Algorithm 2) can be understood as an iteration of our framework. We take as input a labeled train set (X_{train} and Y_{train}) and an unlabeled set (X_{rej}), and two other parameters, η and ρ, to control how many good and bad cases should be added to the training set at this iteration. The parameter η is the total number

Algorithm 1: Retrieve Confident Samples

INPUT : $X_{train}, Y_{train}, X_{rej}, \Delta, c, mode$

```
// 1 - Initialization
```
$X_\Delta \leftarrow \{\}$; $Y_\Delta \leftarrow \{\}$

$N \leftarrow |X_{rej}|$

```
// 2 - Fitting Isolation Forest
```
Fit IsolationForest with X_{train_Δ}

```
// 3(a) - Fitting Classifier
```
if *mode* = *extrapolation* **then**

  Fit Classifier with X_{train}, Y_{train}

```
// 3(b) - Fitting Label Spreading
```
if *mode* = *labelSpreading* **then**

  Fit LabelSpreading with $X_{train}, Y_{train}, X_\Theta$;

```
// 4 - Detect inliers
```
$X_\Theta \leftarrow \{(x_i) \in X_{rej}, outlier(x_i) == False \mid i = 1, ..., N\}$

```
// 5 - Selecting samples
```
while $|X_\Delta| < c$ *and* $|X_{rej}| > 0$ **do**

```
  // 5.1 - Sample identification
```
  $x_j \leftarrow \text{argmax}_j\, P(X_{rej} = \Delta)$

```
  // 5.2 - Sample scoring and evaluation
```
  if $score(x_j) \geq 0.5$ **then**

    $y_j = 1$

  else

    $y_j = 0$

```
  // 5.3 - Adding sample to sets
```
  if $x_j \in X_\Theta$ **then**

    $X_\Delta \leftarrow X_\Delta \cup x_j$

    $Y_\Delta \leftarrow Y_\Delta \cup y_j$

```
  // 5.4 - Removing sample from rejects set
```
  $X_{rej} \leftarrow X_{rej} - \{x_j\}$

OUTPUT: X_Δ, Y_Δ

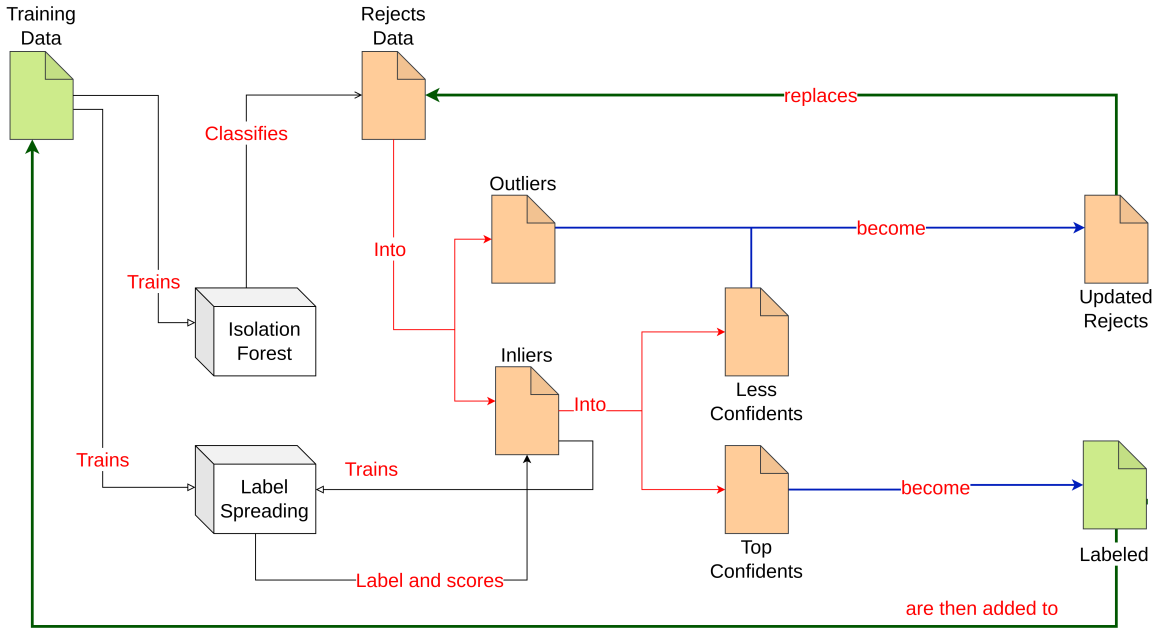

Figure 3.2: Representation of CI-LS framework to perform Reject Inference.

of samples we want to add at this iteration, and the parameter ρ defines the proportion of bad to good payers in the total number of added samples we want to add.

We then call the Retrieve Confident Samples Algorithm (Algorithm 1) within the Expand Dataset Algorithm for both classes, to retrieve c_0 samples with inferred labels for the class *good payers* ($X_{\Delta=0}$, $Y_{\Delta=0}$), and c_1 samples with inferred labels for the class *bad payers* ($X_{\Delta=1}$, $Y_{\Delta=1}$). The samples inferred for both groups are then concatenated to the new training set ($\hat{X}_{train}$ and $\hat{Y}_{train}$) and removed from the unlabeled set, X_{rej}. The expanded training and updated unlabeled sets are returned as the algorithm output.

Algorithm 2: Expand Dataset

INPUT : $X_{train}, Y_{train}, X_{rej}, \eta, \rho$

$c_0 \leftarrow \eta - (\eta \cdot \rho)$;
$c_1 \leftarrow \eta \cdot \rho$;
$X_{\Delta=0}, Y_{\Delta=0} \leftarrow RetrieveTS(X_{train}, Y_{train}, X_{rej}, 0, c_0)$;
$X_{\Delta=1}, Y_{\Delta=1} \leftarrow RetrieveTS(X_{train}, Y_{train}, X_{rej}, 1, c_1)$;
$\hat{X}_{train} \leftarrow Concat(X_{train}, X_{\Delta=0}, X_{\Delta=1})$;
$\hat{Y}_{train} \leftarrow Concat(Y_{train}, Y_{\Delta=0}, Y_{\Delta=1})$;
$\hat{X}_{rej} \leftarrow X_{rej} \setminus \hat{X}_{train}$

OUTPUT: $\hat{X}_{train}, \hat{Y}_{train}, \hat{X}_{rej}$

Chapter 4

Methodology

4.1 Data

We use data from the HomeCredit European dataset (Montoya and KirillOdintsov, 2018) for this research. As well as from the Lending Club dataset (George, 2017), a popular online credit loan platform in the US (Liu et al., 2022a) and used for much research in credit scoring. They are two of the most extensive credit datasets publicly available online. Both datasets were made available on the Kaggle website[1], where competitions related to the identification of bad payers in credit scoring scenarios using these datasets were held.

For the Homecredit dataset, from different files, with different levels of information about the clients' data, we choose to consider only the information present in the *application_train.csv* file for this study. It contains $307,507$ samples from approved clients, with 122 informative features and one target feature. Of the informative features, 106 were numerical, and 21 were categorical. Other files were not considered for this study since they were composed chiefly of information that would only be available for approved clients. This data type would not be useful for us as we are focusing only on the credit granting process.

The Lending Club dataset contains an even more extensive amount of credit data: 2260701 samples for accepted clients and 27648741 samples from rejected clients from 2007 to 2018. Due to this, it can be utilized to train and test a reject inference credit scoring model sufficiently well. This dataset comprises tabular data and contains 151 features for the accepted clients but only nine for the rejected clients. Data from accepted clients can be labeled between *good* and *bad* debtors using the column *Loan_Class*. This data was used to train and test our supervised models. The rejected clients' data is unlabeled and was used to perform Rejected Inference.

4.2 Data pre-processing

Most data pre-processing was made automatically using scikit-learn pipelines (Pedregosa et al., 2011). Due to their structure, which combines several steps of data pre-processing

[1] https://www.kaggle.com/

and classification, they are a helpful tool for data science. By fitting all models inside the pipeline, from processing to classification, with the training data, they also help to avoid data leakage from the testing set. As illustrated in the Figure 4.1 (A), the training data is used to fit the pipeline, and from that, the pipeline can be used to transform and make predictions. When a function is called to predict a testing set, it will apply transformations (pre-processing) to the testing set as necessary based on the values fitted with the training set.

The Figure 4.1 (B) describes the steps implemented on the pipeline created for our experiments. Our pipeline separates features into three categories: numerical Features, categorical Features A, and categorical features B. Group A has categorical features with less than 3 unique values, and group B has at least 3 unique values. Both groups of categorical features are submitted to the same type of null values filling. The mode of the feature is fitted and used to fill the possible missing values of that feature. The null values of the numerical features are filled with the mean instead. Group A is encoded with one hot encoding, which replaces each categorical feature with a column for each unique value. The column will contain the value 1 if the sample has that unique value and the value 0 if it does not. Group B uses a more complex encoding based on Empirical Bayesian Estimation (EBE), avaliable at scikit-learn library as Target Encoder. The Target Encoder replaces categorical values with a value that reflects the proportion of positive cases observed for each category during the fitting process.

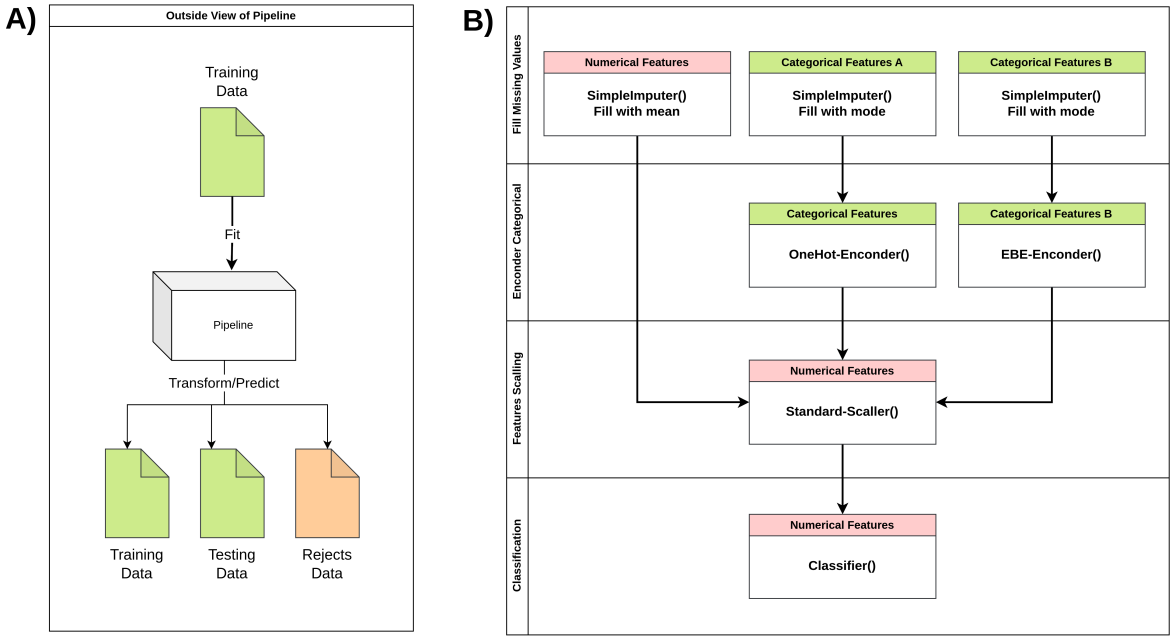

Figure 4.1: (A) Outside view of the pipeline. (B) Inside view of the Pipeline. The pipeline is fitted with the training set and used for pre-processing and classification on all datasets.

4.2.1 HomeCredit

We separated the informative features of the HomeCredit dataset into three subgroups S_1, S_2, and S_3. In S_1, we allocated the features we considered more relevant to the study of

Reject Inference, such as information such as age, number of children, education, and score of a client in other sources, among others, totaling 15 informative features. S_1 features descriptions are listed in Table 4.1[2]. The S_2 subgroup consisted of 71 informative features such as the client's housing situation, the number of times the client's credit information was checked before the loan, and statistics about the building where the client lives. Finally, the S_3 subgroup of features was composed of features like sensitive information such as gender, occupation, and family status of the client, as well as extremely unbalanced features like binary features with information about certain documents, where more than 99% of values were the same for all samples.

Table 4.1: Description of S_1 group of Homecredit features

ID	Features	Description
F1	AMT_CREDIT	Credit amount of the loan
F2,F3 and F4	EXT_SOURCE_1,2 and 3	Normalized score from external data sources
F5	REGION_POPULATION_RELATIVE	Normalized population of region where client lives
F6	DAYS_EMPLOYED	How many days before the application the person started current employment
F7	DAYS_BIRTH	Client's age in days at the time of application
F8	AMT_INCOME_TOTAL	Income of the client
F9	CNT_CHILDREN	Number of children the client has
F10	CNT_FAM_MEMBERS	How many family members does client have
F11	REG_CITY_NOT_WORK_CITY	Flag if client's permanent address does not match work address
F12	AMT_GOODS_PRICE	For consumer loans it is the price of the goods for which the loan is given
F13	ALAG_OWN_CAR	Flag if the client owns a car
F14	NAME_EDUCATION_TYPE	Level of highest education the client achieved
F15	NAME_CONTRACT_TYPE	Identification if loan is cash or revolving
F16	TARGET	Target variable (1 - client with payment difficulties: he/she had late payment more than X days on at least one of the first Y installments of the loan in our sample, 0 - all other cases)

4.2.2 Lending Club

For the Lending Club dataset, we took inspiration from the work of Shih et al. (2022) to make our feature selection for the accepted and rejected clients dataset. However, we decided to avoid certain features in the dataset that would lead to target leaking. These were features related to the credit payment behavior of the client and, thus, were not available to the rejected population. The feature descriptions for the accepted client's dataset are available at Table 4.2. Respectively, Table 4.3 brings the descriptions of the selected features for the rejected clients. The *issue_ d* feature on Table 4.2 and *Application*

[2]The descriptions are provided by the Kaggle repository.

Date on Table 4.3 feature were used to separate the datasets between train and test and were not used to train the models.

Table 4.2: Description of features for accepted clients dataset

ID	Features	Description
A1	addr_state	The state the borrower provides in the loan application.
A2	annual_inc	The self-reported annual income provided by the borrower during registration.
A3	delinq_2yrs	The number of 30+ days past-due incidences of delinquency in the borrower's credit file for the past 2 years.
A4	dti	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.
A5	emp_length	Employment length in years.
A6	home_ownership	The home ownership status provided by the borrower during registration or obtained from the credit report.
A7	int_rate	Interest Rate on the loan.
A8	issue_d	The month which the loan was funded.
A9	last_fico_range_high	The upper boundary range the borrower's last FICO pulled belongs to.
A10	last_fico_range_low	The lower boundary range the borrower's last FICO pulled belongs to.
A11	inq_last_6mths	The number of inquiries in past 6 months (excluding auto and mortgage inquiries).
A12	loan_amnt	The listed amount of the loan applied for by the borrower.
A13	revol_util	Revolving line utilization rate, or the amount of credit the borrower is using relative to all available revolving credit.
A14	term	The number of payments on the loan. Values are in months and can be either 36 or 60.
A15	loan_status	Current status of the loan.

4.3 Evaluation metrics

4.3.1 Area Under the Curve

One evaluation problem in credit scoring is the class imbalance in credit risk datasets. To bypass this problem, metrics such as Area Under the Curve (AUC) are welcomed. AUC is a metric that is not sensitive to threshold values. The higher the AUC value, the better the classifier (Dastile and Celik, 2021). It also reflects the model's performance, even when dealing with unbalanced datasets. The **Area Under the Curve** is given by:

$$AUC = P[p(y = 1|X_i) > p(y = 1|X_j)|y_i = 1, y_j = 0] \tag{4.1}$$

Table 4.3: Description of features for rejected clients dataset

ID	Features	Description
R1	Amount Requested	The total amount requested by the borrower.
R2	Application Date	The date which the borrower applied.
R3	Risk_Score	For applications prior to November 5, 2013, the risk score is the borrower's FICO score. For applications after November 5, 2013, the risk score is the borrower's Vantage score.
R4	Debt-To-Income Ratio	A ratio calculated using the borrower's total monthly debt payments on the total debt obligations, excluding mortgage and the requested LC loan, divided by the borrower's self-reported monthly income.
R5	State	The state provided by the borrower in the loan application.
R6	Employment Length	Employment length in years.

4.3.2 Kickout

Kickout, proposed by Nikita Kozodoi et al. (2019), is a metric that aims to evaluate the performance of a RI model relative to a benchmark model. As illustrated on **??** to calculate this metric we need a labeled test set (from the accepts) and an unlabelled test set (from the rejects). The labeled test set is used to evaluate the benchmark model, and both datasets are used to assess the RI model. It evaluates the number of good and bad cases the model accepts with and without using RI, following the formula in equation 4.2. Considering that we have a benchmark model (BM) without RI, with S_B bad payers, K_B is the number of bad payers accepted by the benchmark model (i.e. false negative cases) now rejected by a method with RI, and K_G is the number of good payers accepted in the benchmark model (i.e. true negative cases) now rejected by a method with RI. $p(B)$ and $1-p(B)$ are the probability of bad and good payers, given that the benchmark model has accepted them. So, $\frac{K_B}{p(B)} - \frac{K_G}{1-p(B)}$ is the difference in the numbers of bad to good payers in proportion to the number of bad and good payers accepted by the benchmark model. And $\frac{S_B}{p(B)}$ is the ratio between the number of ground truth bad payers accepted by the benchmark model and the probability of a ground truth bad payer being accepted by the benchmark model. A good RI model is expected to have a higher kickout value.

$$\text{kickout} = \frac{\frac{K_B}{p(B)} - \frac{K_G}{1-p(B)}}{\frac{S_B}{p(B)}} \tag{4.2}$$

This metric is essential in credit scoring because it can capture the risk of giving credit to bad payers when we include more clients using reject inference. The acceptance rate, α, defines the proportion of clients the models will accept. The decision threshold that separates the clients between accepted and rejected is calculated in the accepted set for the benchmark and in the accepted and rejected set using RI. In the last case, we give credit to more people, and the kickout evaluates how well our exclusion of bad payers went in the RI scenario. A higher kickout reflects a better quality score system when

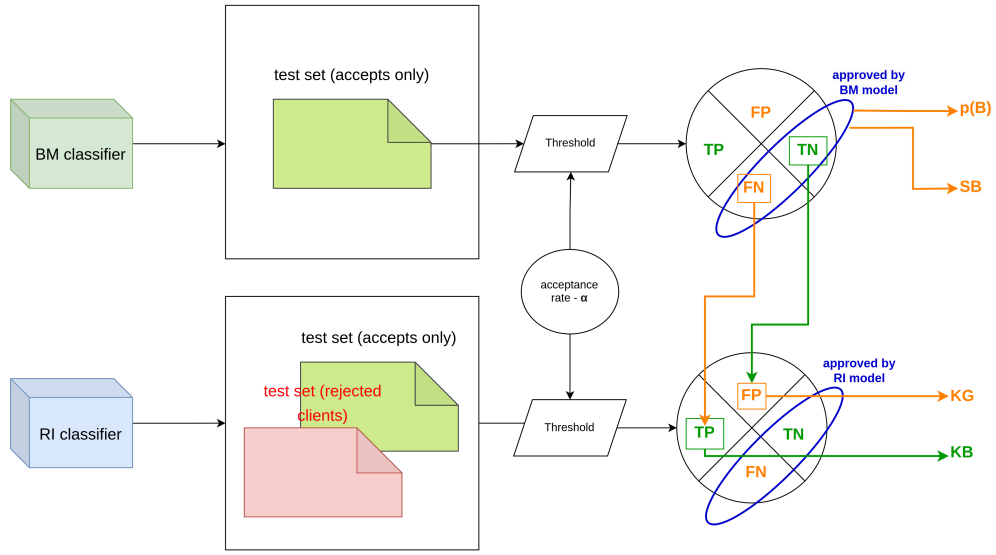

Figure 4.2: Illustration of how the Kickout metric is calculated. TP stands for True Positives, FP for False Positives, TN for True Negatives, and FN for False Negatives.

compared with the benchmark.

The importance of the acceptance rate, α, is worth mentioning. This parameter can create different kickout values depending on its selection. (Nikita Kozodoi et al., 2019) did not explicitly stipulate any value to this variable. For these reasons, we also made a study that evaluated all RI techniques by a range of values for α.

4.3.3 Area Under the Kickout

Our study of how different values of α create an enormous range of kickout values. This leads us to realize focusing on a single value for α may lead to biased conclusions. Therefore, we propose a new metric called Area Under the Kickout (AUK). This metric evaluates the mean of the kickout values for each value of α ranging from 1% to 100%. The formula for calculating the AUK value is given by equation 4.3. In the equation, α represents the percentage of clients the model accepts. The bigger the value of the AUK, the better the model identifies bad clients.

$$\text{AUK} = \frac{\sum_{\alpha=1}^{100} \text{kickout}(\alpha)}{100} \tag{4.3}$$

4.4 Experiment design

We selected the LightGBM (Ke et al., 2017) as our main classifier due to its high performance and faster training speed with tabular data, in particular with the HomeCredit dataset as verified by Daoud (2019). The Label Spreading Algorithm and Isolation Forest implementation came from the Sklearn Library in Python (Pedregosa et al., 2011). We only applied hyper-parameter optimization on LightGBM to the accepted training and validation set. Since all RI techniques tested used the same parameters for the classifier,

we concluded that further parameter tuning was unnecessary.

4.4.1 Experiment I - HomeCredit

In Reject Inference, two types of datasets are necessary — labeled data from the accepted population and unlabeled data from the rejected population. It is essential to compare how employing rejected clients' information will improve the credit scoring system concerning the benchmark model. However, finding public datasets with high levels of information on both accepted and rejected real clients can be pretty challenging. Because of such limitation, taking inspiration from Liu et al. (2022b), we simulated an accept/reject policy using the HomeCredit accepted clients' data to access both an accepted and a rejected data distribution.

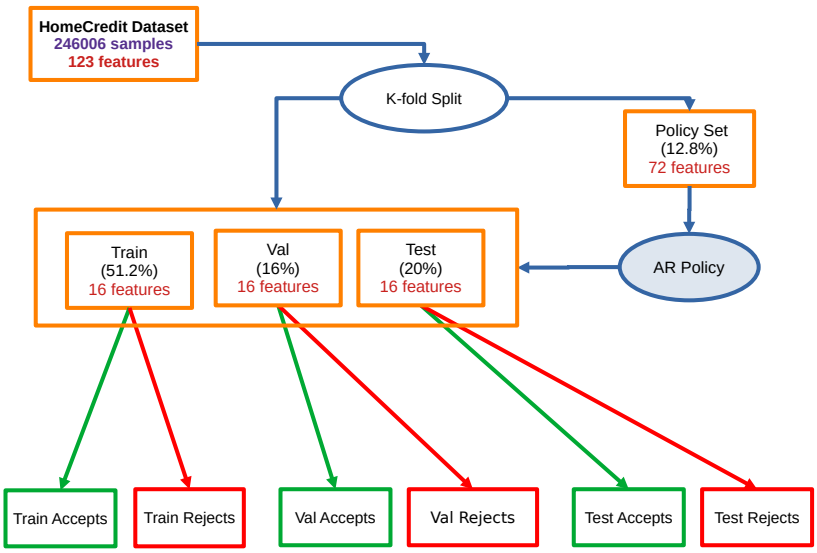

Figure 4.3: The split of the HomeCredit dataset into seven subsets

Figure 4.3 outlines our methodology for splitting the datasets into different subsets of accepted and rejected clients. In this methodology, each reject subset would be considered as unlabeled data. The process of generating each subset starts with the isolation of 20% of samples from the dataset and the cherry-picked features to fit a Logistic Regression classifier. Using Logistic Regression here is inspired by the work of Nikita Kozodoi et al. (2019). According to the authors, this weak learner with L1 regularization is a more reliable way to use the probabilities of default given by the model as a separator between the two classes. We choose $\epsilon = 0.4$ as the threshold value that divides good from risk clients. Whichever sample received a probability of default higher than 0.4 would be allocated to the rejected group of the set. This threshold was chosen to have a large number of rejected individuals, having approximately 1 accepted clients to each 2 rejected ones. Thus following real life scenarios where the number of rejected clients is usually much higher than the number of accepted clients (Nikita Kozodoi et al., 2019; Liu et al., 2022b).

In this experiment, to ensure the robustness of our results, we used K-fold validation to split the training, validation, and test dataset. The simulated accept/reject policy was fitted with 20% of the training set — these samples were ignored temporarily until the next fold split. The proportion of samples for the policy set, as well as for the initials training, testing and validation sets are listed on Figure 4.3. The policy model was applied to each dataset at each K-fold iteration. So for each iteration of the K-fold algorithm, we had seven different subsets, as shown in Table 4.4.

Table 4.4: Description of Dataset Categories

Category	Description
Policy Set	The set used only for fitting the accept/reject policy
Train Accepts	Labeled training set
Train Rejects	Unlabeled training set
Val Accepts	Set used to evaluate the best iteration of our method
Val Rejects	Set used to evaluate the best iteration of our method
Test Accepts	Set used to evaluate all methods
Test Rejects	Set used to evaluate the kickout Metric

Both our techniques produce several versions of progressively bigger training datasets. We choose the TOPSIS method (Chakraborty, 2022) to identify the best version from these datasets that provides the best combination of AUC, with weight 1, and kickout value, with weight 10 according to the stipulated α value. This weight choice was done because we believe that kickout is a more relevant metric than AUC, and in preliminary experiments with validation set, it presented small reduction in AUC.

In this experiment, for our proposed technique, we set η as 1000 and ρ as 0.07, and 0.12 as the value for contamination threshold for the Isolation Forest algorithm. The value of these parameters was obtained through manual fine-tuning.

4.4.2 Experiment II - Lending Club

For the experiments using the Lending Club dataset we choose a different approach. Instead of random K-fold validation, we choose to separate the training and testing sets by time. So for each specific year, the training set is composed of the loans dated from January to September, and the testing set is composed from the loans dated from October to December. However the training and validations sets were created using the widely adopted train and test split function from Scikit-learn library. 70% of the initial training set was kept as training and the remaining 30% used as validation. We followed this protocol for both the accepted and rejected clients datasets. Ultimately, we got six distinct subsets for each year studied. Each subset was used for the same purposes listed in Table 4.4. Figure 4.4 describes the data separation protocol utilized.

Figure 4.4 also lists the years chosen to be analysed in this research, 2009 to 2012, as well as the final number of features selected from each dataset. As mentioned in Section 4.2.2, we took inspiration from the work of Shih et al. (2022) to do the feature selection. We also based our year selection on their work. However, the bigger inspiration

from their work was the data imputation on missing features. As Figure 4.4 there is a different number of features for the rejected and accepted clients datasets, which is a big impediment in machine learning research. Shih et al. (2022) overcomes this issue by applying k-nearest imputation method to fill out the values of missing features on the rejected dataset. This method was proposed by Troyanskaya et al. (2001) and it works by using sample similarity between the accepts training dataset and the rejects datasets to estimate the missing values of the features on the latter, based on the former.

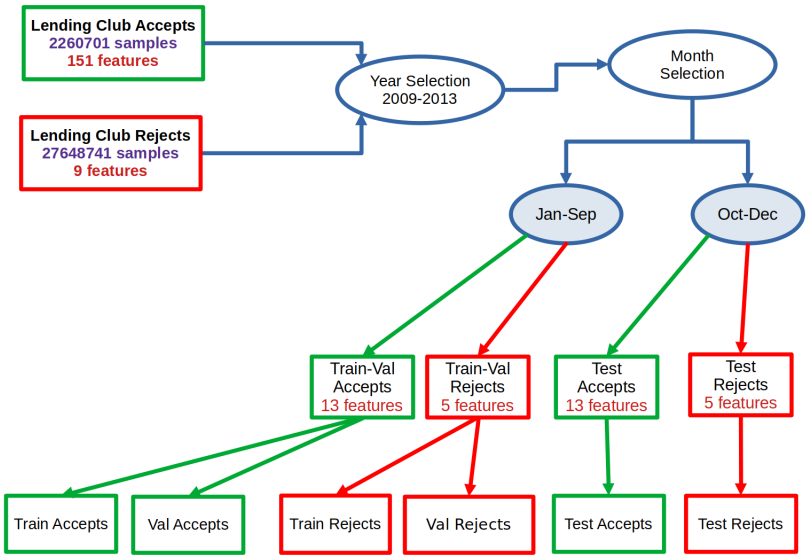

Figure 4.4: The split of the Lending Club dataset into six subsets

Table 4.5: Final Lending Club Dataset Features

	Features	Commentary
Present on both datasets	R1 and A12	Same Feature.
	R2 and A8	Same Feature. Used for dataset splitting.
	R3 and A9-A10	Same Feature. The mean values of A9 and A10 were used.
	R4 and A4	Same Feature.
	R5 and A1	Same Feature.
	R6 and A5	Same Feature.
Present only on Accepts	A2, A3, A6, A7, A11, A13, A14	Values on Rejects were filled with KNN inputation.
	A15	Used as target feature.

Table 4.5 describes the final version of the features utilized for this experiment. As mentioned in the table, Instead of using both features A9 and A10 from the accepts dataset, we choose to use the mean of these two features. By doing this, the resulting feature represents the same information of feature R3, risk score, on the rejects dataset. With the exception of the feature A15, which was used as the target feature of our experiment, all the features present only on the accepts dataset were filled on the rejects

dataset with the values obtained using the k-means imputation from scikit-learn. In the end, for this experiment we had 12 informative features and 1 target feature to train our models.

In this experiment, for our proposed technique, we set η as 1000 and ρ as 0.2, and 0.12 as the value for contamination threshold for the Isolation Forest algorithm. The value of these parameters was obtained through manual fine-tuning. For this experiment we also choose the TOPSIS method to identify the best version from the resulting datasets obtained by our proposed techniques. However, instead of making the selection using kickout values, we used the AUK metric, which is less biased towards a acceptance rate value.

Chapter 5

Results and Discussion

Given the distinct methodology made for both the Homecredit and Lending Club datasets, we will analyse their experiment results individually. However, for both experiments all the classification steps of both ours and the compared techniques were made using a different instance of the Lightgbm model with the same parameters and seed number. We compared our proposed frameworks with the techniques previously mentioned in the Section 2.2.1 and Section 2.2.2.

- **BM - Benchmark model**: Model trained with only data from accepted set;
- **A-SC - Augmentation with Soft Cut-Off**: Weight Adjusting and Data Inflating Method;
- **A-UW - Upward Augmentation**: Weight Adjusting Method;
- **A-DW**[1] **- Downward Augmentation**: Weight Adjusting Method;
- **A-FU - Fuzzy-Augmentation**: Weight Adjusting Method;
- **E-C - Confident Extrapolation**: Data Inflating Method;
- **PAR - Parcelling**: Data Inflating Method;
- **LSP - Label Spreading**: Data Inflating Method.
- **CI-EX - Confident-Inlier Extrapolation for Reject Inference** : Data Inflating Method;
- **CI-LS - Confident-Inlier Label Spreading for Reject Inference**: Data Inflating Method.

5.1 Results Using Simulated Rejected Clients

For the experiment using the Homecredit dataset, with simulated rejected clients, we evaluated the performance of our models through three aspects, which included comparing

[1]Only used for experiment with the HomeCredit dataset.

the kickout Value at the common threshold α of 50%[2], exploring the kickout value for multiples values of α, and the trade-off between AUC and kickout metrics.

The Figure 5.1 presents the mean kickout value for all techniques on 5 K-fold validation when using an acceptance rate of 50%. All the values of kickout for the BM model are zero because it is compared with itself when calculating this metric. From this figure, we can observe that Data Inflating Methods generally lead to higher values of kickout. Although we had contrasting results between the E-C and the LSP technique, both simpler versions of the ones we proposed, which the Figure 5.1 shows, have the best performance in this metric.

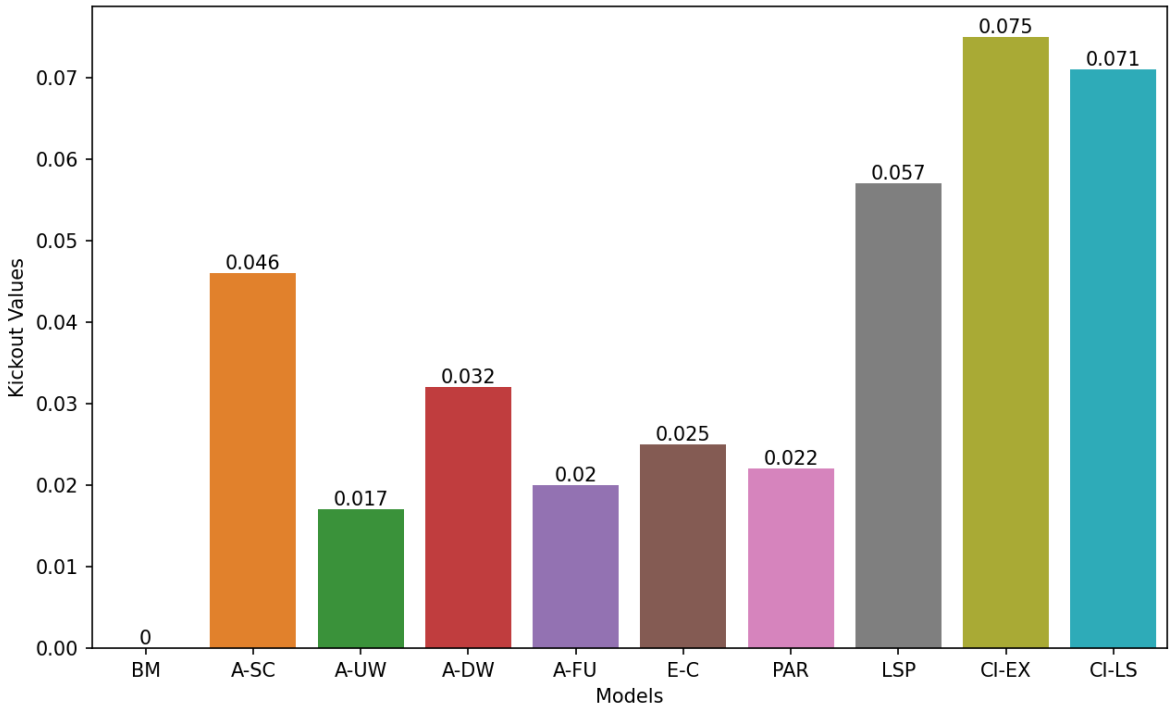

Figure 5.1: Comparing kickout value at an acceptance rate of 50% for all techniques. The common classification threshold for machine learning models.

Figure 5.2 shows our proposed methods slightly decrease AUC performance when compared with the base model (BM). However, the AUC loss is less than 2 percent points even when compared to the BM model. This figure shows our proposed methods (and other RI techniques) do not improve AUC value for the accepted population nor decrease this value significantly. In this scenario, the capability of the proposed method of improving the kickout measure is quite important, allowing a more qualified inclusion of rejected clients.

During our experiments, we observed the value of kickout is heavily influenced by the acceptance rate α. Therefore, we present in the graph of figure 5.3 the evolution of kickout metric with changing acceptance rates. We can observe a clear growth tendency for the kickout values for all techniques when we increase the value of α. The second conclusion from this graph is that our proposed techniques, CI-EX and CI-LS, frequently offer the

[2]We featured this value for comparison, with the assumption this a safe value for separate between good and bad payers.

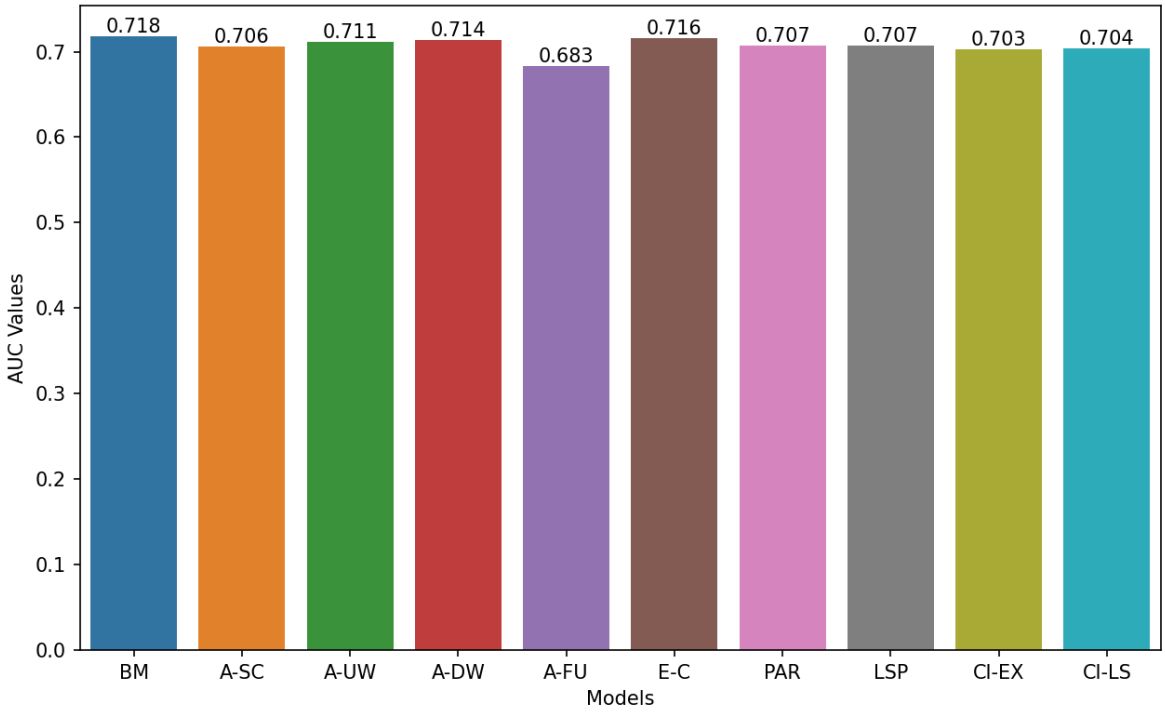

Figure 5.2: Comparison of average AUC values (5 K-fold).

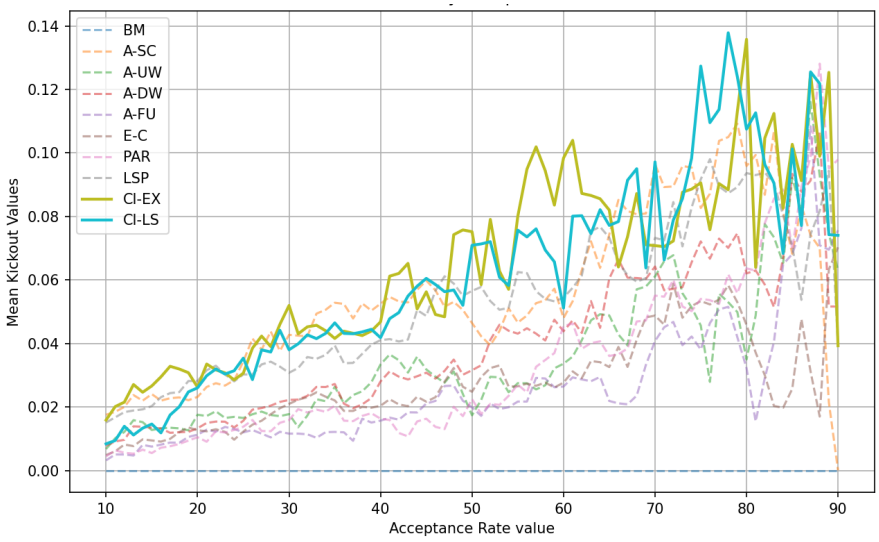

Figure 5.3: Comparing evolution of kickout value by acceptance rate for all techniques (5-fold cross-validation).

highest kickout values regardless of the stipulated acceptance rate. The CI-EX technique, in particular, is more indicated for strict credit policies when the rejection rate desired is higher.

The AUC value is an important credit scoring metric. However, it has strong limitations in the task of evaluating RI techniques since, in most cases, it can only evaluate the accepted population due to the lack of labels of the rejected clients, and RI techniques aim at improving the classification of the whole population (Kozodoi et al., 2020). For this reason, even though the RI techniques did not offer higher results of AUC than the BM model, as shown in Figure 5.2, we argue the kickout results shown in Figure 5.3, demonstrate the ability of RI metrics at correctly identifying bad cases in comparison with the BM model, creating better quality credit scoring policies. Finally, by observing the slight AUC decrease with a robust and consistent increase in kickout, we can say that the proposed method is a competent RI method that can improve credit scoring quality.

5.2 Results Using Real Rejected Clients

For the experiment using the Lending Club dataset, with real rejected clients, we evaluated the performance of our models through three aspects, which included comparing the AUC of the RI methods versus the model with only accepts, comparing the kickout value of the RI models at the common threshold α of 50%[3], and finally exploring the mean of kickout values of the RI models for a complete range of percentages of acceptance rate with the AUK metric. The final results for these metrics can be found in Table 5.1 and in the following figures.

Figure 5.4 shows a graph of how the models performed based in the AUC metric for the years 2009 to 2012. As can be seen in the figure, for most years, the BM model offers the highest perform in AUC compared to the RI models. The A-SC and A-UW got very good AUC values, almost matching the BM model, and in one case even surpassing the BM model. From our proposed models, CI-EX and CI-LS, only in the year 2011, CI-EX surpassed the BM model. However, for most cases it offered competitive results. The same can not be said about our CI-LS model, which along with the LS model, both models used label spreading, were usually one of the models with the worst AUC values.

Figure 5.5 shows another graph of how the models performed for the years 2009 to 2012, but this time based in the metric kickout with 50% acceptance rate. The results showed in this figure express a more clear advantage of our proposed methods, CI-EX and CI-LS, compared to the others RI techniques studied. It can be seen in the figure that for this acceptance rate, the kickout value of most RI techniques is equal to 0. This means that these techniques are not identifying any bad cases that the BM model did not already identify. The only time methods others than our proposed ones present any value of kickout different than zero was the year 2009, which was the year with the least amount of samples for all datasets. Except for the year 2009, our proposed method CI-LS was the model with the highest kickout value, being followed by our other method CI-EX

[3]Again we featured this value with the assumption this a safe value for separate between good and bad payers.

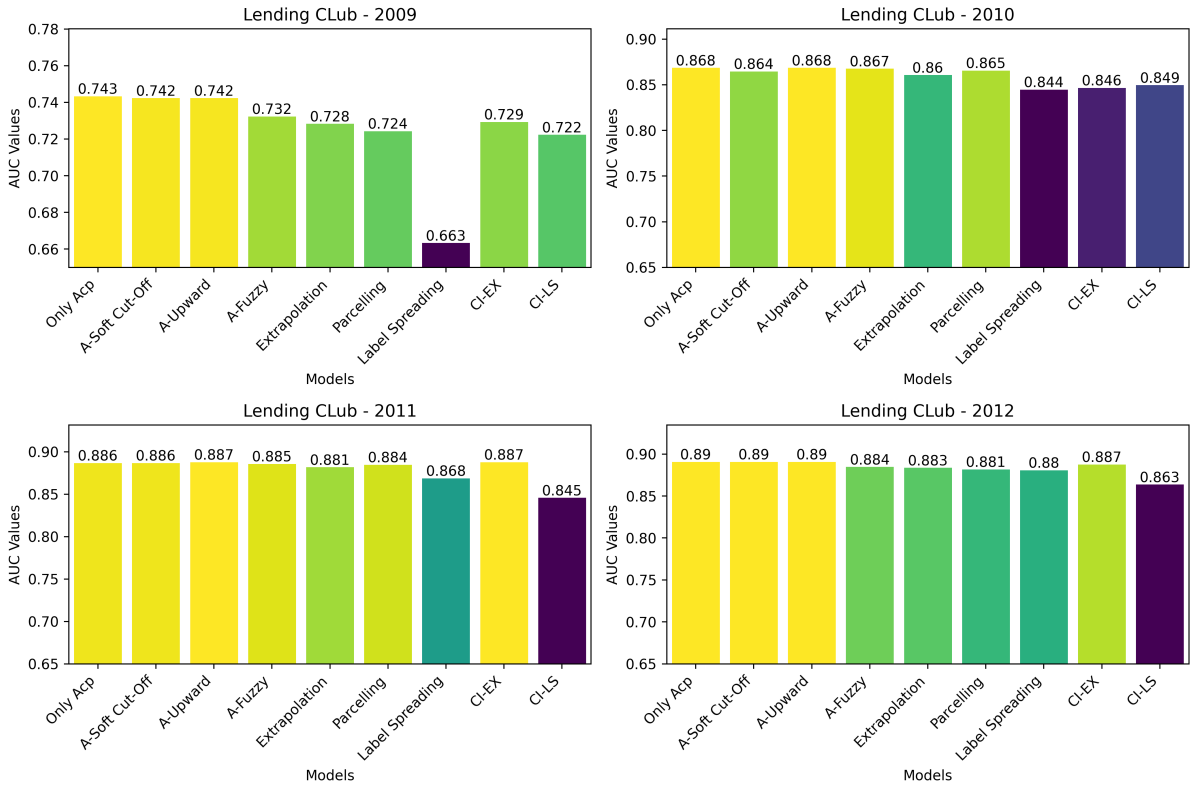

Figure 5.4: Mean results for AUC metric of the studied models from 5 experiments with different seeds. Lending Club dataset from years 2009 to 2012.

in the years 2010 and 2012.

Figure 5.6 shows a more detailed view of how the RI models perform at the task of correctly avoiding approving loan for bad cases in proportion to avoiding incorrectly denying loans to good cases. In the figure, displayed in blue are the models that had a positive value of AUK, that means they were, in mean, better than the BM model at kicking out bad cases. Displayed in red, are the models with negative values of AUK, which means they were, in mean, worst than the BM model at denying loans to good cases, without actually avoiding enough bad cases. It can be seen in the graphs the models A-FU, E-C, and PAR had negative values for all years studied. In contrast, our model, CI-LS had positive values, as well as the highest values, for all years studied. Our other model, CI-EX, as well as the LS model, only had a result lower than zero for the year 2011.

As mentioned at the end of Section 5.1, the AUC metric alone can not identify the best RI model. Yet, the kickout metric is neither perfect. Because it depends on a parameter, α, that does not have a unique indicated value for all scenarios, it is hard to ascertain its reflection in practical application results. And although we could not verify in real world applications our proposed AUK metric, we argue it is a strong indicative of how well the studied RI models would rank in a real world scenario. As it is seen in the graphs, and in the Table 5.1, RI models with highest AUC values in most cases did not achieve positive performance in neither kickout or AUK metrics. In contrary, the models with the lowest AUC were the ones that had consistent good values for AUK. However, our proposed

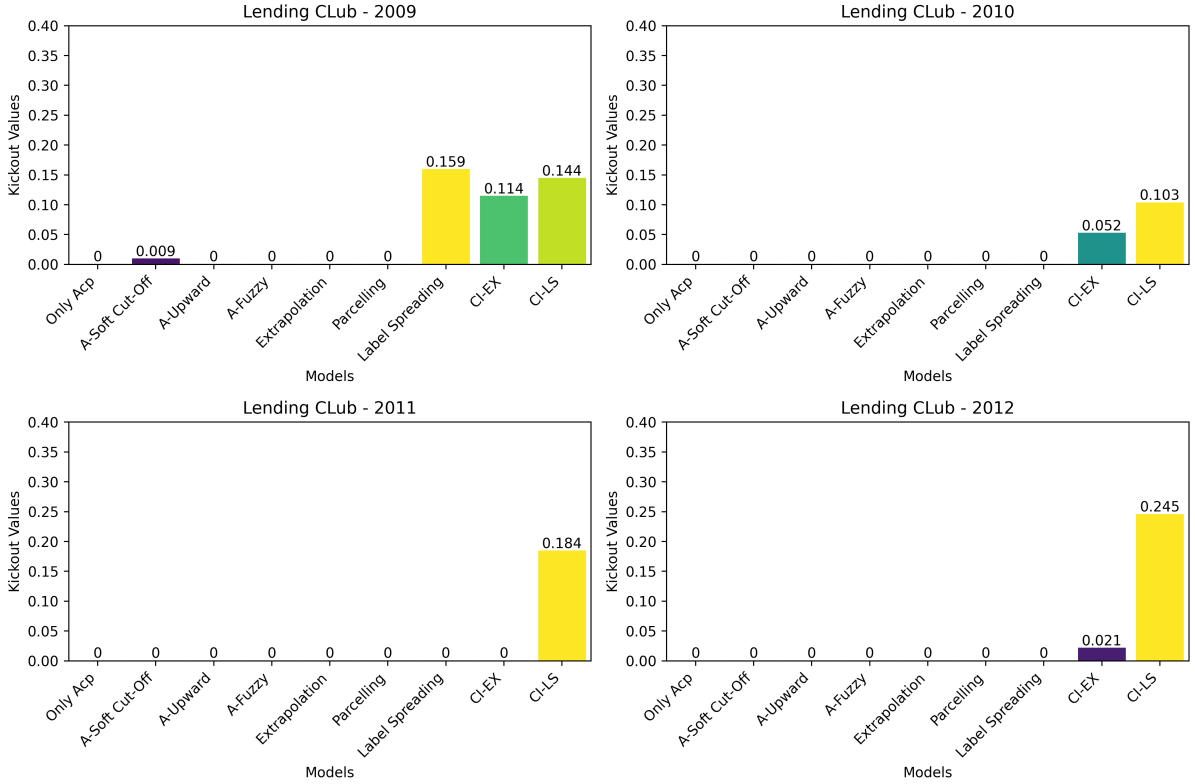

Figure 5.5: Mean results for kickout metric ($\alpha = 0.5$) of the studied models from 5 experiments with different seeds. Lending Club dataset from years 2009 to 2012.

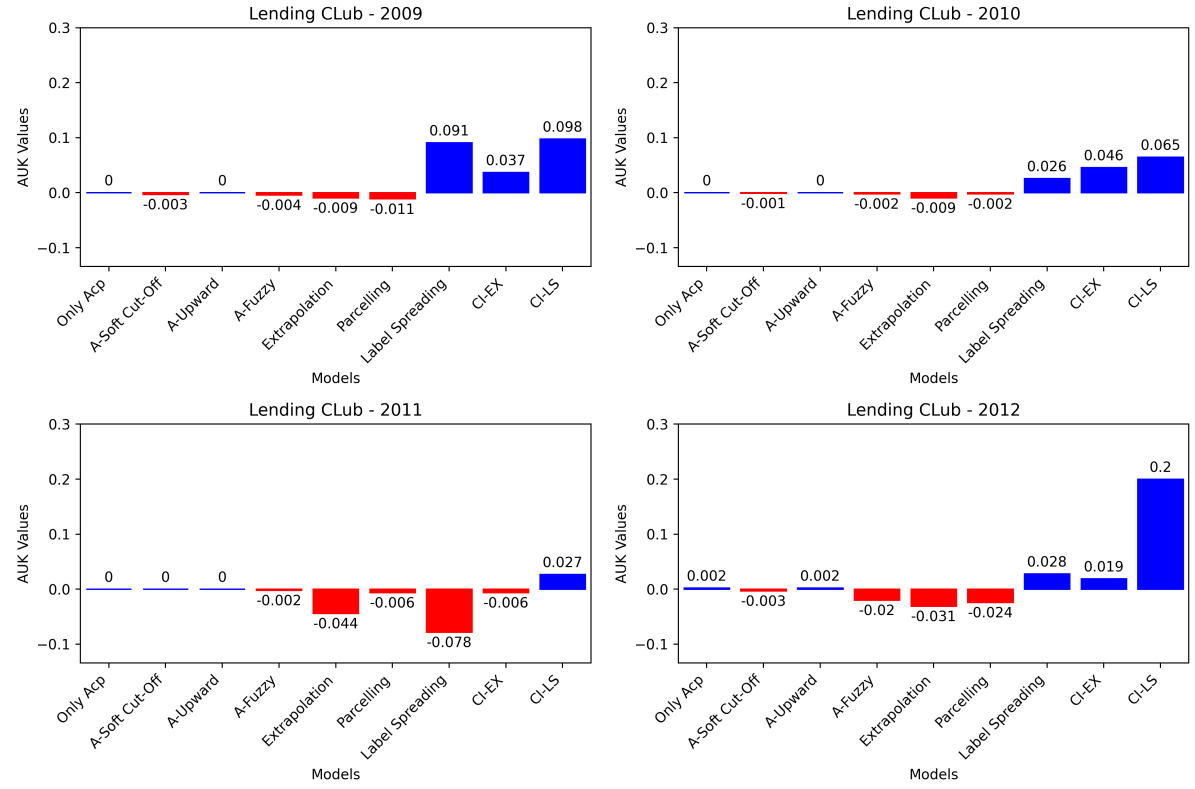

Figure 5.6: Mean results for AUK metric of the studied models from 5 experiments with different seeds. Lending Club dataset from years 2009 to 2012.

model CI-EX, for most years studied, is the exception. The CI-EX model was able to achieve positive results for AUK and kickout without losing much AUC.

Both our models, due to their iterative nature, are much more time consuming to train than the other studied models. And the CI-LS model is even more slow at training than our other proposed CI-EX model. This seems to indicate to achieve higher AUK values it is necessary to compromise some AUC value as well as invest more training time. But this may be worth considering how most studied models with high AUC achieve negative AUK values, raising the question of whether they were really of any improvement to the BM model.

Table 5.1: Comparison of Model Metrics

Year	Metrics	BM	A-SC	A-UW	A-FU	E-C	PAR	LSP	CI-EX	CI-LS
2009	AUC	**0.743**	0.742	0.742	0.732	0.728	0.724	0.663	0.729	0.722
	Kickout	-	0.009	-	-	-	-	**0.159**	0.114	0.144
	AUK	-	-0.003	-	-0.004	-0.009	-0.011	0.091	0.037	**0.098**
2010	AUC	**0.868**	0.864	**0.868**	0.867	0.860	0.865	0.844	0.846	0.849
	Kickout	-	–	-	-	-	-	-	0.052	**0.103**
	AUK	-	-0.001	-	-0.002	-0.009	-0.002	0.026	0.046	0.**065**
2011	AUC	0.886	0.886	**0.887**	0.885	0.881	0.884	0.868	**0.887**	0.845
	Kickout	-	-	-	-	-	-	-	-	**0.184**
	AUK	-	-	-	-0.002	-0.044	-0.006	-0.078	-0.006	**0.027**
2012	AUC	**0.890**	**0.890**	**0.890**	0.884	0.883	0.881	0.880	0.887	0.863
	Kickout	-	-	-	-	-	-	-	0.021	**0.245**
	AUK	0.002	-0.003	-	-0.020	-0.031	-0.024	0.028	0.019	**0.200**

Chapter 6

Conclusions and future work

This research proposes two novel semi-supervised frameworks for Reject Inference. They are both variations of a confident inlier approach proposed by this work. In this approach, we apply outlier detection as a filter to pre-select samples closer to each class distribution and confident criteria to make a rigid selection of samples from those. However, while the CI-EX variation uses the probabilities of a classifier trained with the accepted samples to label rejects, the CI-LS variation labels the rejected population using semi-supervised learning with the label spreading algorithm. Using two large public datasets, we compare our proposed methods with relevant RI techniques from the literature. We use two literature performance criteria, AUC and kickout, and a novel metric for RI, AUK. Using the HomeCredit dataset, the results of the proposed methods offered the highest predictive power concerning the kickout metric for all acceptance rates tested without a significant AUC loss. Using the Lending Club dataset, our proposed methods achieve more contrasting results than the other RI models from the literature, with some more significant loss in AUC in some cases but with consistently good results for kickout and AUK metrics.

The results of this work support the usefulness of outlier detection and semi-supervised learning in Reject Inference, as well as the importance of looking at Reject Inference from a different perspective. To the best of our knowledge, this is the first paper evaluating classical Reject Inference techniques with kickout, a much less unbiased metric for this type of technique. It is one of the few works that propose a new way of evaluating RI methods in a less biased way. We conclude from these experiments that even classical techniques can improve the baseline model (trained with only accepts), challenging the conclusions of Crook and Banasik (2004).

However, our work has some limitations. First, our proposed frameworks take longer to train than the compared methods. Second, we only tested the RI models with one classifier model. Future work should aim to improve the computational cost of our algorithms, test different models and strategies for filtering confident samples, and increase the rate of incorporating rejected data into the training set. A relevant way to continue this work is to use this framework on Brazilian credit datasets. We would also like to investigate new metrics for validating the positive impact of RI approaches on marginalized populations and improving the RI metric already proposed in this work.

6.1 Ethics Statement

All data used in this study were anonymized and sourced from publicly available datasets, ensuring that no personally identifiable information was included or processed. We recognize the importance of fairness and transparency in predictive modeling, especially in financial services where decisions can significantly impact individuals' lives. However, we did not submit our models to fairness metrics to guarantee they did not favor nor harm any particular group, although we plan to address this in future work. We view our work not as the final step towards a fair credit scoring system, but as one of many steps necessary to achieve it. We plan to release our code publicly to facilitate transparency and reproducibility of our results.

Bibliography

Muhammad Ali, Peimin Zhu, Ma Huolin, Heping Pan, Khizar Abbas, Umar Ashraf, Jar Ullah, Ren Jiang, and Hao Zhang. A novel machine learning approach for detecting outliers, rebuilding well logs, and enhancing reservoir characterization. *Natural Resources Research*, 32(3):1047–1066, 2023.

Raymond A Anderson. *Credit Intelligence and Modelling: Many Paths Through the Forest of Credit Rating and Scoring*. Oxford University Press, 2022.

Subrata Chakraborty. TOPSIS and Modified TOPSIS: A comparative analysis. *Decision Analytics Journal*, 2:100021, March 2022. ISSN 2772-6622. URL `https://www.sciencedirect.com/science/article/pii/S277266222100014X`.

Lize Coenen, Ahmed K. A. Abdullah, and Tias Guns. Probability of default estimation, with a reject option. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*, pages 439–448, October 2020. doi: 10.1109/DSAA49011.2020.00058. URL `https://ieeexplore.ieee.org/abstract/document/9260038`.

J Crook and J Banasik. Does reject inference really improve the performance of application scoring models? *Journal of Banking & Finance*, 28(4):857–874, April 2004. ISSN 03784266.

Essam Al Daoud. Comparison between XGBoost, LightGBM and CatBoost Using a Home Credit Dataset. *International Journal of Computer and Information Engineering*, 13(1):6–10, January 2019. URL `https://publications.waset.org/10009954/comparison-between-xgboost-lightgbm-and-catboost-using-a-home-credit-dataset`.

Xolani Dastile and Turgay Celik. Making Deep Learning-Based Predictions for Credit Scoring Explainable. *IEEE Access*, 9:50426–50440, 2021. ISSN 21693536.

Adrien Ehrhardt, Christophe Biernacki, Vincent Vandewalle, Adrien Ehrhardt, Christophe Biernacki, Vincent Vandewalle, Philippe Heinrich, and Sébastien Beben. Reject Inference Methods in Credit Scoring : A rational review To cite this version : HAL Id : hal-03087279 Reject Inference Methods in Credit Scoring :. *Journal of Applied Statistics*, 2020.

Monir El Annas, Badreddine Benyacoub, and Mohamed Ouzineb. Semi-supervised adapted hmms for p2p credit scoring systems with reject inference. *Computational Statistics*, pages 1–21, 2022.

Nathan George. Lending Club Loan Data. Kaggle, 2017. https://www.kaggle.com/datasets/wordsforthewise/lending-club.

Zhiyu Guo, Xiang Ao, and Qing He. Transductive semi-supervised metric network for reject inference in credit scoring. *IEEE Transactions on Computational Social Systems*, 2023.

Yanzhe Kang, Ning Jia, Runbang Cui, and Jiang Deng. A graph-based semi-supervised reject inference framework considering imbalanced data distribution for consumer credit scoring. *Applied Soft Computing*, 105:107259, 2021.

Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

Nikita Kozodoi, Panagiotis Katsas, Stefan Lessmann, Luis Moreira-Matias, and Konstantinos Papakonstantinou. Shallow self-learning for reject inference in credit scoring. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2019, Würzburg, Germany, September 16–20, 2019, Proceedings, Part III*, pages 516–532. Springer, 2020.

Jingxian Liao, Wei Wang, Jason Xue, and Anthony Lei. Data Augmentation Methods for Reject Inference in Credit Risk Models. 2021.

Jingxian Liao, Wei Wang, Jason Xue, Anthony Lei, Xue Han, and Kun Lu. Combating Sampling Bias: A Self-Training Method in Credit Risk Models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12566–12572, June 2022. ISSN 2374-3468. URL `https://ojs.aaai.org/index.php/AAAI/article/view/21528`. Number: 11.

Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*, volume 793. John Wiley & Sons, 2019.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

Jiaming Liu, Sicheng Zhang, and Haoyue Fan. A two-stage hybrid credit risk prediction model based on XGBoost and graph-based deep neural network. *Expert Systems with Applications*, 195:116624, jun 2022a. ISSN 0957-4174.

Qiang Liu, Yingtao Luo, Shu Wu, Zhen Zhang, Xiangnan Yue, Hong Jin, and Liang Wang. Rmt-net: Reject-aware multi-task network for modeling missing-not-at-random data in financial credit scoring. *IEEE Transactions on Knowledge and Data Engineering*, 2022b.

Loretta J Mester et al. What's the point of credit scoring. *Business review*, 3(Sep/Oct): 3–16, 1997.

Anna Montoya and Martin Kotek KirillOdintsov. Home Credit Default Risk. Kaggle, 2018. https://kaggle.com/competitions/home-credit-default-risk.

Nikita Kozodoi, P. Katsas, S. Lessmann, L. Moreira-Matias, and Konstantinos Papakonstantinou. Shallow Self-Learning for Reject Inference in Credit Scoring. 2019. URL https://www.semanticscholar.org/reader/62e4d60277138d15eebf0b47e22deb8cd002f6b1.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Gabriele Sabato. SOLVING SAMPLE SELECTION BIAS IN CREDIT SCORING: THE REJECT INFERENCE. n.d.

Feng Shen, Xingchao Zhao, and Gang Kou. Three-stage reject inference learning framework for credit scoring using unsupervised transfer learning and three-way decision theory. *Decision Support Systems*, 137:113366, 2020.

Dong-Her Shih, Ting-Wei Wu, Po-Yuan Shih, Nai-An Lu, and Ming-Hung Shih. A Framework of Global Credit-Scoring Modeling Using Outlier Detection and Machine Learning in a P2P Lending Platform. *Mathematics*, 10(13):2282, January 2022. ISSN 2227-7390. URL https://www.mdpi.com/2227-7390/10/13/2282. Number: 13 Publisher: Multidisciplinary Digital Publishing Institute.

Naeem Siddiqi. *Intelligent credit scoring: Building and implementing better credit risk scorecards*. John Wiley & Sons, 2017. ISBN 978-1-119-27915-0.

Mengnan Song, Jiasong Wang, and Suisui Su. Towards a better microcredit decision. *arXiv preprint arXiv:2209.07574*, 2022.

Olga Troyanskaya, Michael Cantor, Gavin Sherlock, Pat Brown, Trevor Hastie, Robert Tibshirani, David Botstein, and Russ B Altman. Missing value estimation methods for dna microarrays. *Bioinformatics*, 17(6):520–525, 2001.

Yufei Xia. A Novel Reject Inference Model Using Outlier Detection and Gradient Boosting Technique in Peer-to-Peer Lending. *IEEE Access*, 7:92893–92907, 2019. ISSN 2169-3536. Conference Name: IEEE Access.

Dengyong Zhou, Olivier Bousquet, Thomas Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. *Advances in neural information processing systems*, 16, 2003.