Women's World Banking

Tackling Credit Fairness and Missed Business Opportunities with REJECT INFERENCE

Mehrdad (Mehi) Mirpourian

April 2024

Contents

Tables and Figures

Acknowledgments

This report is the culmination of the dedicated efforts and unwavering support of many esteemed colleagues and organizations.

I am grateful for the invaluable contributions of peers across the industry. My sincere thanks go to Dr. Charalampos Chelmis, Sadia Rahman, and Mahsa Azarshab at the University at Albany, State University of New York, whose research into robust and counterfactual learning has been pivotal in addressing the complexities of reject inference. I also offer my deepest appreciation to Dr. Sonja Kelly and Dr. Megan Dwyer Baumann, whose consistent support and guidance were indispensable in the progression and success of this project. Their expertise and encouragement have been a guiding light throughout this journey. Lastly, I extend my profound gratitude to PayPal. Their generous support provided essential resources for intellectual freedom, paving the way for innovative research and exploration in this field. PayPal's steadfast dedication to advancing financial inclusion and promoting fair finance has been a pillar in the development of this research. Their commitment has not only provided a solid foundation for this project but also inspired a deeper understanding and exploration of equitable financial practices, enriching the impact and relevance of our findings.

Executive Summary

Access to fair and affordable credit is vital for fostering economic growth and enhancing financial inclusion, and it offers particular benefits for lower-income households and individuals.

People seek credit for a variety of needs, including personal, business, and educational purposes. Financial institutions evaluate credit applications using algorithms, loan officers, or a combination of both. However, these credit evaluation methods are not infallible but are susceptible to errors arising from biases or assessment mistakes. Consequently, many potentially creditworthy applicants are erroneously rejected, resulting in lost opportunities for both the individuals and the financial institutions.

Reject inference is a quantitative approach that helps institutions to understand and infer the reasons behind these erroneous rejections, and to identify applicants who were mistakenly deemed non-creditworthy. These methods are applied towards the end of the credit evaluation process, and they offer a non-disruptive solution to financial institutions, since the methods do not require major changes to existing credit assessment procedures.

In this paper, I delve into the application of data science and artificial intelligence (AI) in developing innovative reject inference algorithms. I present two main categories of algorithms, each uniquely tailored according to the nature of the data and the specific reasons for credit rejections. The first category comprises algorithms based on matching techniques, known for their intuitiveness, relative ease of implementation, and effectiveness in identifying applicants who have been mistakenly rejected. The second category builds upon a cutting-edge noisy label detection and correction algorithm, tailored specifically for the reject inference problem. This method employes advanced AI techniques to identify erroneous rejections. The algorithms in both categories possess unique strengths, making each suitable for different scenarios in reject inference. The choice of method depends on the specific nature of the data and the underlying reasons for credit rejections. Those factors will determine the most effective option for each situation.

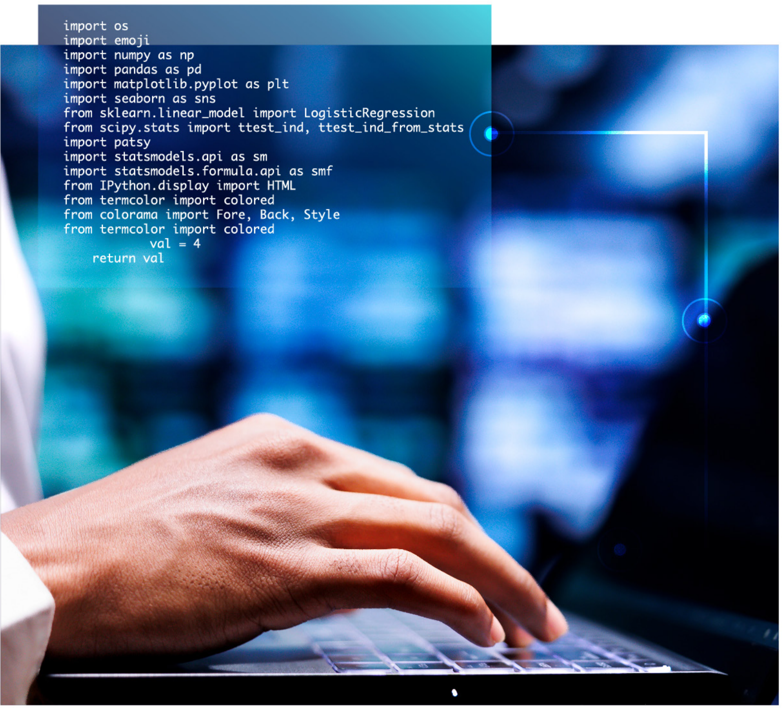
Women's World Banking initiated this project with the aim of applying the knowledge and insights it has gathered about the application of reject inference to address gender biases against rejected applicants, as detailed in the works of Mirpourian et al. (2023) and Kelly and Mirpourian (2021). The objective was to develop a comprehensive educational framework that draws from extensive field experiences and equips practitioners with the necessary tools and understanding to apply these methods effectively and thoroughly.

Introduction

This study investigates unfair credit rejections by analyzing the use of reject inference methods. It emphasizes the ethical and economic consequences of erroneous credit application rejections and discusses how machine learning (ML) can be utilized to reduce detrimental effects.

The report has two main sections: The first delves into the ethical and business implications of unfair credit rejections and provides conceptual insights; the second part details two methodologies used to identify and address unjust credit application rejections, with an in-depth examination of each technique.

In the development and selection of the technical approaches showcased in this research, I have given particular emphasis to their practical applicability for financial services providers (FSPs). This focus on real-world utility stems from Women's World Banking extensive field experience and hands-on work with datasets from financial institutions. Such engagement has allowed me to fine-tune the methods to make them directly relevant and useful for practitioners. The innovative methods presented in the second section of this report are not just theoretical constructs, but are driven by practical insights from the field, and grounded in solid data science techniques.



```
import os
import emoji
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from scipy.stats import ttest_ind, ttest_ind_from_stats
import patsy
import statsmodels.api as sm
import statsmodels.formula.api as smf
from IPython.display import HTML
from termcolor import colored
from colorama import Fore, Back, Style
from termcolor import colored
            val = 4
    return val
```

This paper forms a part of a broader initiative designed to introduce the reject inference technique to the financial practitioner's community. To complement this report, we have created an online course detailing the methods discussed here. This course is available on the Women's World Banking YouTube channel and its official webpage, and is narrated by Mehi Mirpourian, data science manager at Women's World Banking, and Dr. Charalampos Chelmis, Associate Professor in the Computer Science Department at the University at Albany, State University of New York.

Furthermore, we have developed a Python toolkit is available on Women's World Banking's official GitHub page. These resources are crafted to enable technical experts to implement the reject inference techniques outlined in this report.

With these extensive resources–including, the online course, the Python toolkit, and our empirical research—our objective is to share the full scope of the project with industry practitioners. We are committed to promoting fair lending practices and to underscoring not only the ethical and economic benefits of such practices to economies, but also the potential of fair lending to enhance financial inclusion.

Section I

FOUNDATIONAL CONCEPTS

The following foundational concepts are essential to any project that examines reject inference in credit applications. The succinct explanations of the concepts are intended to provide enough context for readers new to the topic to effectively understand and apply reject inference techniques.

Credit Assessment

Credit assessment is the evaluation of the creditworthiness of an individual, business, or entity. The assessment aims to ascertain that entity's capacity to meet financial obligations, especially in the context of borrowing or extending credit. A credit assessment process includes the analysis of diverse financial and non-financial factors in order to gauge the risk associated with lending money or providing credit. FSPs may employ algorithms, loan officers, or a hybrid model combining both, in an effort to assess creditworthiness.

Gender Bias in Credit Assessment

Gender bias in credit assessment refers to the existence of discriminatory elements, influenced by gender, in the process of evaluating an individual's creditworthiness. Gender bias may result in the unequal treatment of an individual during credit approval, interest rate determination, or other credit-related terms, even when that person's financial profile is comparable to someone who did not experience discrimination.

Sources of Gender Bias in Credit Evaluation

Kelly and Mirpourian (2021) identified the following three main categories of sources of bias:

HISTORICAL DATA

If the historical data used for training credit algorithms reflects past biases, such as gender-based lending disparities, then current algorithms may perpetuate the biases by learning from the historical patterns.

MODEL DEVELOPMENT

Biases can emerge during the design phase of credit algorithms (algorithmic design), when the developers, whether intentionally or unintentionally, introduce criteria that favor or disadvantage specific gender groups.

HUMAN BIASES

Human judgment in credit assessment can be influenced by implicit biases. Loan officers might unintentionally favor, or disfavor, applicants based on gender-related stereotypes, impacting credit approval or rejection decisions.

Ethical Implications of Gender Bias in Credit Evaluation

Gender bias in credit assessment has profound ethical consequences, affecting individuals, society, and the financial industry. These consequences include the perpetuation of unfair treatment, reinforcement of harmful stereotypes, financial injustices, deepening social and economic inequalities, erosion of trust in financial institutions, potential legal and regulatory challenges, hindrance of financial inclusion efforts, and customer dissatisfaction. Addressing gender bias is an ethical imperative and is crucial for fostering a financial system that upholds fairness, equality, and the dignity of all individuals, irrespective of gender.

Financial Impacts of Gender Bias in Credit Evaluation Processes

Gender bias in credit assessment can have detrimental business and financial consequences. Financial institutions may lose market share as customers seek fairer alternatives. Institutions may also experience reputational damage leading to decreased trust and loyalty, incur legal and regulatory penalties for engaging in discriminatory practices, and miss out on business opportunities due to limited economic participation.

Algorithmic Solutions to Gender Bias in Credit Assessments

Overall, there are three primary algorithmic solutions for addressing algorithmic bias: preprocessing methods, in-processing methods, and post-processing methods. Preprocessing methods are applied during the data preparation stage—before the model training phase—with the goal of preprocessing or manipulating input data to address biases or imbalances before the algorithm learns patterns from it. In-processing methods are integrated directly into the model training process, modifying the learning algorithm or training the process itself to reduce biases or ensure fairness. Post-processing methods are applied after the model has been trained. Post-processing methods seek to mitigate biases in the model's predictions or outcomes by adjusting its outputs.

In this report, I will focus exclusively on studying reject inference. In the context of this paper, reject inference can be categorized as a post-processing technique. Readers interested in preprocessing and in-processing methods can explore them through additional resources. Shelke et al. (2017) have listed different upsampling and downsampling methods that can be used as a preprocessing technique, and Celis et al. (2020) conducted research on the use and impact of a maximum entropy-based approach for data preprocessing. Zhang et al. (2018) proposed a method of using adversarial learning to mitigate unwanted bias, providing insights into in-processing techniques. For post-processing techniques, readers can refer to the fairness-aware ensemble framework developed by Losifidis et al. (2019).

At this stage, the reader should have gained sufficient familiarity with credit assessment, the manifestation of gender bias within it, consequences of bias in credit assessment, and the fundamental strategies to address gender bias. With this foundational knowledge, the discussion can now transition to the specific area of reject inference.

KEY CONCEPTS IN CREDIT REJECT INFERENCE

Reject inference is a technique used to predict or estimate outcomes for cases that were excluded from a specific system or model. These are cases for which we lack performance data because they were not included. The aim is to determine their potential outcomes if they had been included from the start. In the context of credit assessment, reject inference involves making predictions about the creditworthiness or risk associated with applicants who were denied credit. Reject inference attempts to leverage the available data and estimate what the outcome would have been for those rejected credit applicants. Reject inference is valuable for detecting and mitigating biases in credit assessment models, since it helps to uncover potential disparities in the treatment of various groups—different genders.

In this report, the examination of reject inference stems from three key insights gained from Women's World Banking's practical experience with various FSPs globally.

1 Reject inference has the potential to mitigate the adverse consequences of the amplified bias effect.

In the credit offering sector, the concept of a feedback loop, or the amplified bias effect, plays a significant role. A feedback loop occurs when the outcomes of a process are used as inputs for the same process in future iterations, which often ends up reinforcing the initial bias or error. This is particularly evident in cases where initial credit rejections, stemming from biases or errors, negatively impact an individual's credit history. Such rejections can create a self-reinforcing cycle in which the affected applicants face increased difficulties in obtaining credit in the future, due to their now-worsened credit histories. This is where the reject inference becomes vital. It helps in pinpointing those individuals who, despite their initial rejection, are likely to be creditworthy. By correctly identifying these cases, reject inference effectively counters the amplified bias effect, leading to more equitable and precise credit evaluations.

2 Reject inference can enhance the credit assessment processes used by FSPs, without necessitating major alterations to their existing credit evaluation practices.

Many methods have been devised for credit assessment. In our collaborations with more than 10 different FSPs over their credit assessment methods, we have noted that altering an FSP’s credit assessment practice often meets with considerable resistance. Consequently, persuading an FSP to change its established credit assessment procedures to enhance fairness and avoid rejecting potentially creditworthy applicants can be an immensely challenging, if not impossible, task. However, our advocacy suggests that it is important not to abandon the pursuit of fairness due to the considerable difficulties involved in altering standardized credit assessment methods. Instead, we champion the use of reject inference. This approach can be integrated effortlessly with existing credit assessment methods, thus not disrupting established practices. The implementation of reject inference by institutions represents a practical first step towards greater fairness. It encourages a gradual shift towards more equitable assessment processes, offering a realistic path to improving fairness in credit evaluations.

3 In markets that are heavily saturated with the presence of FSPs, acquiring new customers poses a challenge, and erroneously rejecting potential customers can lead to increased costs.

In Women's World Banking's collaborations with multiple FSPs, we have observed that offering credit in highly competitive markets presents its own specific challenges. The digital credit landscape is densely populated with a plethora of active applications, complicating the tasks of customer acquisition and retention. Erroneous rejections, which mistakenly deny credit to worthy individuals and lead to the inadvertent loss of potential customers, have even more damaging implications in such competitive environments.

A BRIEF OVERVIEW OF REJECT INFERENCE METHODS IN CREDIT SCORING

Reject inference in credit scoring has historically relied on classical statistical methods, as seen in Hand and Henley's (1997) exploration of reject inference techniques and in Banasik and Crook's (2007) focus on augmentation and reweighting. However, despite the foundational nature of reject inference methods, they have often hinged on potentially unrepresentative assumptions about unobserved data. The introduction of Bayesian statistical methods in credit scoring, as discussed by Hand (2001), added a probabilistic dimension to reject inference and enriched the spectrum of analytical approaches. The parallel evolution of ML techniques transformed reject inference methods. Li et al. (2017) demonstrated the efficacy of semi-supervised support vector machines (SVMs) for reject inference problems, and Shen et al. (2020) applied transfer learning to enhance credit scoring, introducing new ML paradigms to reject inference. Additionally, researchers like Mancisidor et al. (2020) have explored deep generative models, presenting advanced solutions for improving credit scoring accuracy. The evolution of reject inference has not been limited to modeling techniques. The emergence of fintech companies underscored the need for alternative data in credit assessment, as discussed by Mitra et al. (2023). The shift toward the use of alternative data highlights the growing relevance of diverse data sources in financial applications. Still, despite extensive research in reject inference, there remains a disconnect between academic studies and real-world applicability in financial institutions. This report aims to bridge these gaps by presenting new perspectives on reject inference, aimed specifically at practitioners. In what follows, we present two reject inference methods. First, we explore propensity score matching technique as a suitable method because of its conceptual alignment with reject inference. This method is not only aligned with the principles of reject inference but is also easily understandable, even for those without a strong technical background. Second, we examine the potential of treating reject inference as a classification problem with noisy labels. Viewing mistakenly rejected cases as mislabeled data points opens the door to more robust decision-making in certain scenarios, despite the complexity involved. In the following sections, we delve into each method and discuss its applicability and its advantages in various scenarios.

Section II

ALGORITHMIC METHODS TO TACKLE REJECT INFERENCE

This section introduces two distinct approaches to the implementation of reject inference. The first one leverages matching techniques. The second approach employs counterfactual learning, focusing on classification in environments with noisy labels. The matching-based approach is straightforward to both understand and to implement. This method is suitable for scenarios where FSPs anticipate significant human biases during the final stages of their credit evaluation processes. Conversely, the second approach effectively addresses situations influenced by human biases, as well as cases in which rejections are not exclusively due to such biases.

APPROACH I

Addressing Reject Inference with ML Based Propensity Score Matching Techniques

To tackle reject inference through matching algorithms, this section outlines a strategy that integrates Propensity Score Matching (PSM) from causal inference with ML techniques, thereby enhancing PSM's effectiveness. The section begins with a concise overview of PSM, followed by an explanation of its application in reject inference. The section then delves into incorporating ML techniques within the PSM framework. It concludes with a discussion of how to evaluate the performance of the PSM algorithm.

The PSM Algorithm and its Application in Reject Inference Effect

Matching methods, including PSM, are a class of non-parametric approaches that take observational data and match individuals with similar characteristics but different treatments, in order to make causal inferences. The intuitive nature of PSM, along with its statistical benefits, renders it a valuable tool for reject inference. PSM creates a statistical comparison group based on a model that predicts the likelihood of receiving the treatment (e.g. a loan in credit applications), using observable characteristics. This probability, known as the propensity score (PS), is then used to match participants (loan recipients) with non-participants (rejected applicants). The efficacy of PSM hinges on two key conditions. The first is conditional independence, which implies that all relevant decision-making variables by an FSP are known, and no hidden factors influence the outcome. The second is sufficient overlap in PSs between the approved and rejected groups, ensuring viable comparisons.

Implementation Steps of PSM in Reject Inference

To address the reject inference problem through PSM, I have designed an algorithm that comprises 14 steps. This section starts with a brief overview of these steps, as outlines in Table 1, then delves into a comprehensive explanation of each.

Reject Inference via PSM Algorithm

TABLE 1. THE 14 STEPS OF REJECT INFERENCE VIA PSM ALGORITHM

STEP NUMBER	DESCRIPTION
1	Divide the dataset. Randomly select 80% of the approved cases as the training set, reserving the remaining 20% as the test set.
2	Compute PSs for both approved and rejected individuals using a chosen algorithm, such as logistic regression.
3	Validate the quality of the calculated PSs.
4	Select a metric to determine the adequacy of pairings between approved and rejected cases based on their PSs.
5	Attempt to match each rejected applicant with approved cases, using the chosen metric and the difference between the PSs of the rejected applicant and the approved cases.
6	Decide on a matching strategy, such as one-to-one or one-to-many matches, based on the metric from Step 5. I suggest using both strategies.
7	Exclude any rejected applicants for whom a match cannot be found, given the metric and the difference in PSs.
8	If a single match is found for a rejected applicant, evaluate their default status, and record the loan performance of the approved match.
9	If multiple matches exist for a rejected applicant, use the majority vote on loan performance from the approved matches.
10	Repeat the process for each rejected applicant until all have been matched.
11	Assess the performance of your algorithm using different matching strategies (e.g. one-to-one, one-to-three, etc.), using the test data to determine the most effective approach. ** A more detailed explanation of the right performance metric is provided when Step 11 is explained in detail.*
12	Repeat Steps 2 - 11 using other algorithms, including SVMs, Random Forest, XGBoost, Multi-layer Perceptron (MLP) Neural Network, Generalized Additive Models (GAM), and K-Nearest Neighbors (KNNs).
13	Compile the results into a single table.
14	Select the algorithm and matching strategy that performs best on the test set.

What follows is a detailed explanation of each of the 14 steps above, and an elaboration of the methodologies employed.

STEP 1

A distinctive characteristic of this algorithm is its approach to measuring PSM's performance. I employ a test dataset, distinct from the training dataset, to assess the effectiveness of the matching algorithm. This step goes beyond the usual statistical assumptions of PSM, aligning with common practices in ML algorithm development. I suggest an 80:20 data split between training and testing, although other ratios or cross-validation methods can also be employed. Notably, this data split applies only to the approved applicants. Further details on utilizing the test set will be elaborated in Step 14. For now, it is important to note that that 80% of the approved cases will form the training set and the remaining 20% will form the test set.

NOTE

When dealing with longitudinal data or data with a temporal dimension, a random split may not be appropriate. For longitudinal data, where observations are collected from the same subjects over time, the best approach to sampling involves strategies that maintain the integrity of the temporal order and account for within-subject correlations. Therefore, other sampling strategies such as stratified sampling can be the right strategy.

STEP 2

A pivotal step in this algorithm is the calculation of PSs for both approved and rejected credit applicants. PS represents the probability of an individual receiving a treatment, given that person's observed characteristics. In the context of credit scoring, "treatment" refers to the approval of a loan. To calculate the PSs, I utilize different ML algorithms to model the relationship between a binary treatment variable (in this case, loan approval or rejection) and a set of observed characteristics of the applicants, such as income level, credit history, and other relevant financial indicators. Each applicant, whether approved or rejected, is assigned a PS to indicate that individual's likelihood of being approved for a loan based on their profile. These scores are necessary for the next steps, where they serve as a basis for matching rejected applicants with approved ones who have similar scores. By pairing applicants in this way, I aim to assess whether rejected individuals might have been creditworthy, based on the loan performance of their approved counterparts who share similar characteristics.

While logistic regression is commonly employed in PSM studies to calculate PS, it may not always yield the optimal results. Given the varied nature and distribution of real-world data, other methods might surpass logistic regression in effectiveness. In ML, a standard approach is to develop a diverse array of relevant models, then select the one that demonstrates the best performance on a test set. Consequently, this study will outline a range of ML algorithms deemed suitable for calculating PSs. More details about the implementation of these methods can be found on the official Women's World Banking GitHub page associated with this study. Here, as part of our elaboration of Step 2, we give an overview of each of these methods and how they can be applied to calculate the PSs.

Logistic Regression for PS Estimation

Logistic regression operates by estimating the odds of an event occurring. For each applicant, it computes a score (log of odds) between 0 and 1, based on the characteristics of the applicant. This score reflects the likelihood of an applicant being approved for a loan as per the logistic model. Essentially, the output of the logistic regression model for each individual in our dataset is the same as the applicant's PS.

SVMs for PS Estimation

SVM is a powerful ML tool used for classification and regression problems. In the context of this study, SVM is employed to distinguish between two groups: those who have been approved for a loan and those who have been rejected. Unlike logistic regression, which directly models the odds of loan approval, SVM works by finding the optimal boundary that separates approved and rejected applicants. This is achieved by identifying and leveraging a set of key data points, known as support vectors, which are critical in defining the decision boundary in the data space.

Once the SVM model is trained on the dataset, it categorizes applicants as either likely to be approved or likely to be rejected based on their attributes. The distance of an applicant's data point from the decision boundary can be interpreted as a propensity score. This score reflects the certainty with which the model assigns an applicant to the approved or rejected category, akin to the probability score in logistic regression. If an applicant's data point is far from the boundary, it suggests a higher confidence in the classification. For an approved applicant, this means the model is quite certain about that individual's approval; for a rejected applicant, the model is equally certain about the rejection. If an applicant's data point is close to the boundary, it indicates lower confidence in the classification. This could imply that the applicant's case is more ambiguous, and the model is less certain about whether that individual should be approved or rejected. To translate this into a propensity score-like measure, one might consider the relative distance from the boundary. An applicant closer to the boundary could be seen as having a more moderate PS, indicating a less clear-cut decision, while applicants further away would have higher or lower PSs, indicating clearer decisions. However, this interpretation requires careful calibration and understanding of the SVM model's output.

Random Forest for PS Estimation

Random Forest, an ensemble ML method, offers a robust approach for calculating PSs in reject inference. This method builds numerous decision trees to classify each applicant as either approved or rejected for a loan. In Random Forest, the PS is derived from the proportion of trees that classify an applicant as approved. For example, if 80 out of 100 trees predict approval for an applicant, the PS would be 0.8. This score reflects the likelihood of loan approval based on the applicant's profile. Random Forest's approach, which considers complex interactions among variables, provides a dynamic alternative for PS estimation.

XGBoost for PS Estimation

XGBoost, a highly efficient implementation of gradient boosted trees, is another method for computing PSs in reject inference studies. Renowned for its performance and speed, XGBoost builds an ensemble of decision trees sequentially, with each tree correcting the errors of its predecessors. In the context of calculating propensity scores, XGBoost classifies applicants into the approved or rejected category. The PS for each applicant is essentially the predicted probability of loan approval and is generated by the ensemble of trees. This probability is derived from the aggregated predictions of all the trees in the model. Due to its powerful handling of large and complex datasets, and its ability to model non-linear relationships, XGBoost provides a nuanced and accurate method for estimating PSs.

✓ MLP Neural Network for PS Estimation

An MLP Neural Network, a type of deep learning model, can be employed for PS calculation in reject inference. MLPs consist of multiple layers of neurons, with each layer fully connected to the next, allowing for the modeling of complex and non-linear relationships. In the context of PS estimation, an MLP is trained to classify applicants into two categories: those likely to be approved for a loan and those likely to be rejected. The network learns from a set of input features to make these predictions. The output of the MLP, typically from a sigmoid activation function in the final layer for binary classification, gives a probability score. The probability score represents the likelihood of an applicant being approved for a loan based on the individual's characteristics. In essence, the probability score serves as the PS, quantifying each applicant's likelihood of loan approval as determined by the neural network. The advantage of using an MLP for this task lies in its ability to capture complex patterns in the data, which might be missed by more traditional models. However, MLPs require large datasets for training and often lack interpretability.

✓ GAM for PS Estimation

GAM offers a flexible approach to calculating PSs in reject inference studies. GAM extends linear models by allowing non-linear relationships between the independent variables and the dependent variables using smooth functions. In applying GAM for PS estimation, the model assesses the likelihood of loan approval for each applicant. This is achieved by analyzing various applicant attributes and understanding their non-linear impact on the approval decision. The output of a GAM in this context is the probability of an applicant being approved for a loan, considering that individual's unique combination of characteristics. This probability effectively serves as the PS, indicating how likely an applicant is to receive a loan based on the model's assessment. The advantage of GAMs is their capability to discern complex relationships and patterns in the data while maintaining the interpretability of the model outputs. This is significant because, typically, models that capture intricate relationships tend to sacrifice interpretability, making them unfavorable to use for inference. However, GAMs stand out by offering both sophistication in analysis and clarity in understanding.

✓ KNNs for PS Estimation

KNN, a straightforward yet powerful ML algorithm, can be utilized to estimate PS in reject inference. KNN works on the principle of similarity, identifying the 'k' closest data points (neighbors) to a given data point—and making predictions based on these neighbors. In the context of PS calculation, KNN classifies each credit applicant as either likely to be approved or likely to be rejected. This classification is based on the proximity of the applicant to others in the dataset. The PS in KNN is inferred from the proportion of the nearest neighbors that are approved applicants. For example, if an applicant's closest neighbors are predominantly approved, this indicates a higher likelihood of loan approval. KNN's conceptual simplicity and reliance on actual data points for classification make it a valuable tool for calculating PS.

TABLE 2. COMPARISON OF MACHINE LEARNING METHODS IN PROPENSITY SCORE ESTIMATION

METHOD	PROS	CONS	BEST USED WITH
Logistic Regression	Well-understood, interpretable, good for linear relationships	May not capture complex, non-linear relationships	Small to medium datasets, simpler relationships
SVM	Effective in high-dimensional spaces, robust to outliers	Less interpretable, can be computationally intensive	Data with clear margins of separation, moderate to large datasets
Random Forest	Handles large datasets well, good for non-linear data, less prone to overfitting	Can be complex to interpret, requires careful tuning	Large and complex datasets with many features
XGBoost	Highly efficient, effective with large and complex datasets	Can be prone to overfitting, requires tuning	Large and complex datasets when performance is a priority
MLP Neural Network	Captures complex patterns, effective for non-linear relationships	Black-box model, requires large amounts of data	Large datasets with complex, non-linear patterns
GAM	Flexible, can model non-linear relationships, interpretable	Can struggle with very large datasets, requires selection of smooth functions	Datasets where relationship modeling is important, medium-sized datasets
KNN	Simple and intuitive, effective in scenarios with clear clusters	Sensitive to the choice of 'k', struggles with high dimensionality	Data with clear clusters, smaller datasets

STEP 3

In the realm of reject inference, the validation of PSs is a critical step, particularly given the diversity in dataset complexities. Two widely adopted approaches for this validation are Balance Checks and Visualization Techniques.[1]

✓ Balance Checks

This approach ensures that the PSM effectively balances the covariates between the treated (approved applicants) and control (rejected applicants) groups. Statistical tests such as t-tests are employed to evaluate the differences in covariates between groups. A small or statistically insignificant difference indicates the success of PS in creating comparable groups, thus validating the groups' effectiveness. This method is particularly crucial as it directly measures the PS model's ability to create equivalent groups, which is central to the integrity of this analysis.

[1] A comprehensive guide for implementing the PS validation is available on the project's GitHub page.

Visualization Techniques

Given the complexity of datasets and the nuanced nature of reject inference, visual inspection of PS distributions can provide immediate, intuitive insights. Techniques such as plotting histograms of PS or creating love plots (which show the balance of covariates before and after matching) offer clear visual representations of the overlap and distribution of scores. Visualizations are especially useful in detecting any overt imbalances or anomalies in the distribution of scores, thereby complementing the statistical rigor of balance checks with an accessible, interpretable overview.

STEP 4

Once the PSs have been validated, the subsequent phase in reject inference analysis is to conduct the matching process between approved and rejected applicants. The matching process requires establishing a metric or criteria to ascertain the degree of similarity necessary for two PSs to be deemed a match. Various approaches can be employed, such as nearest-neighbor matching, caliper matching, and radius matching. Nearest-neighbor matching involves pairing each rejected applicant with an approved applicant whose PS is the closest. In caliper matching, a maximum permissible difference, or caliper, is set between PSs. Matches are identified only if they fall within this specified range. Radius matching defines a range or radius around the PS of each rejected applicant, within which matches are identified.

NOTE

In the approach and code used in this paper, I have used caliper matching.

STEP 5

Compute the absolute difference in PSs between each rejected applicant and all approved cases. Retain only those pairings where the difference in scores meets the criteria set by your chosen metric from Step 4.

STEP 6

In this step, you will decide about the matching strategy, with options ranging from one-to-one to one-to-many matches. This choice will be guided by the metric established in the previous step. While it is best to use both strategies, the focus in this paper is primarily on one-to-many matches, specifically targeting odd numbers up to a maximum of 15 matches per rejected applicant.

As explained in Step 5, for each rejected applicant, it is necessary to first identify all approved cases that meet the matching criteria. If only one approved case meets the matching criteria for a rejected applicant, it results in a one-to-one match. If more than one approved case satisfies the criteria, proceed to one-to-many matching. However, to maintain a manageable and analytically sound approach, limit the number of matches to odd numbers, with a maximum of 15. In scenarios where the potential matches exceed this limit (e.g. 20 matches), you can select the top 15 matches based on the smallest absolute differences in PSs.

STEP 7

If no matches are found for a rejected applicant, you can drop that applicant and discard it from the analysis.

```
df_sum_stat_f = df_female.describe()
    df_sum_stat_m = df_male.describe()
    df_sum_stat = pd.concat([df_sum_stat_f, df_sum_stat_m], keys =
["Women", "Men"])
    # dropping the unnecessary rows to make the table more readable
    df_sum_stat.drop(index = ('Women', 'count'), inplace = True)
    df_sum_stat.drop(index = ('Women', 'min'), inplace = True)
    df_sum_stat.drop(index = ('Women', 'max'), inplace = True)
    df_sum_stat.drop(index = ('Men', 'count'), inplace = True)
    df_sum_stat.drop(index = ('Men', 'min'), inplace = True)
    df_sum_stat.drop(index = ('Men', 'max'), inplace = True)
    # printing the summary stats table
    return df_sum_stat
```

```
def missing_stat(df_f, df_m):
    print(colored('Percentage of missing values in each column - broken down by gender:', 'blue',
attrs = ['bold']))
    print()
    f_miss = (np.round(df_f.isna().sum()/len(df_f),2)).to_frame(name = '# % of missing - Women
applicants')
    m_miss = (np.round(df_m.isna().sum()/len(df_m),2)).to_frame(name = '# % of missing - Men
applicants')
    missing_data = pd.concat([f_miss, m_miss], axis = 1)
    print(missing_data)
```

STEPS 8-9

After successfully identifying the best matches for each rejected applicant using the approach explained in Step 6, the next move is to estimate the loan performance of the rejected applicants who were paired with approved individuals. To estimate the rejected individuals' loan performance, you can adopt a binary representation of loan performance: '1' indicates an applicant who has fully repaid the loan without defaulting, while '0' represents an applicant who has defaulted. This analysis is conducted through a majority voting system. For instance, consider a scenario where three approved applicants are matched to a rejected case. We assess the loan performance of these three individuals. If two out of the three have successfully repaid their loans (non-defaulters), the calculation would be as follows: (0+1+1)/3=0.66. The result signifies that 66% of the approved matches for this rejected applicant have repaid their loans. Therefore, based on our method, we can estimate that the rejected applicant had a 66% likelihood of repaying the loan. This majority vote approach provides a numerical value representing the estimated probability of loan repayment and offers a data-driven basis to infer the potential creditworthiness of rejected applicants.

NOTE

The rationale behind recommending an odd number of matched cases stems from our adoption of the majority vote approach. An odd number of pairings ensures a definitive majority, facilitating a clear decision in the voting process.

NOTE

Based on the analysis in this paper, consider a scenario with three matched pairs yielding a 66% reliability rate. Notably, this outcome could arise with varying numbers of pairs, such as 15. Reliability improves with more matches, but traditionally, institutions would set a minimum match threshold based on risk tolerance. Our approach, however, differs because we assess algorithm performance and the best matching strategy based on using test data. This data-driven method allows the test set to dictate the optimal matching strategy, ensuring a more empirical and accurate analysis.

STEP 10

The steps of the chosen algorithm, as detailed for a single rejected applicant, must be systematically applied to all rejected applicants. This involves repeating Steps 1 through 9 for each individual in the rejected category.

STEP 11

This step focuses on evaluating the selected algorithm's performance and determining the most effective matching strategy for predicting the creditworthiness of rejected applicants. Utilizing the test set, which contains 20% of approved cases with known loan outcomes, we apply our algorithm to identify potential matches from the training set for each individual. We then use various matching strategies (such as 1-to-1, 1-to-3, etc.) to predict loan performance through a majority vote system. This process is repeated across the entire test set. The accuracy of our predictions is verified against the actual loan performance data. We calculate the error rate for each matching strategy, reflecting the proportion of individuals within the test set whose loan performance was incorrectly predicted. The final stage involves comparing these error rates to select the matching strategy that had the most success, thereby determining the most reliable approach for assessing the creditworthiness of rejected applicants. This step ensures the selection of a strategy that enhances the overall accuracy of our reject inference analysis.

NOTE

In this analysis, we employed the train-test approach to assess the performance of our matching algorithms, then tailored the code accordingly. Alternatively, cross-validation, such as 5-fold or 10-fold, can be utilized to determine the optimal matching strategy.

NOTE

As previously noted, our algorithm is particularly relevant in scenarios where an FSP anticipates biases from loan officers, or when it is using datasets historically influenced by such biases. This forms the foundational assumption and enables us to evaluate our method's performance on test data. Essentially, this assumption relies on the lack of systematic differences between certain segments of rejected and approved applicants. It allows us to presume that the test set mirrors the distribution of some segments of the rejected applicants. This similarity becomes significant in instances where loan officers may reject creditworthy applicants based solely on subjective, prejudiced decisions against certain individuals or groups.

STEP 12

Repeat Steps 2 through 11 using the other algorithms suggested in this study. This process ensures a comprehensive assessment of various methods.

STEP 13

Document the error rates provided by each algorithm and matching strategy when applied to the test set. This step makes it possible to compare the effectiveness of different approaches.

STEP 14

Identify and select the algorithm and matching strategy that demonstrated the best performance based on the error rates observed in the test set. This choice will be critical in ensuring the highest accuracy and efficiency of the reject inference analysis.

APPROACH II

Addressing Reject Inference Using Noisy Label Detection and Counterfactual Correction

This section explores an algorithm that was initially designed for noisy label correction and examines how this method can be adapted and effectively utilized within the realm of reject inference, followed by an explanation of its application in reject inference. The section then delves into incorporating ML techniques within the PSM framework. It concludes with a discussion of how to evaluate the performance of the PSM algorithm.

Key Terms and Concepts

Before delving into the noisy label detection algorithm and its application in reject inference, it is essential to familiarize yourself with some key terms and concepts that will be frequently referenced throughout this discussion. I will start with a summary of the concepts used in this approach, followed by an introduction to the main method.

Labeled Dataset

Labeled data refers to a dataset in which each entry is tagged with a label that represents specific attributes or characteristics that are necessary for the operation of learning algorithms. In such datasets, every individual item is marked with a particular label or outcome. For example, consider a dataset of credit users in which applicants are categorized as either "defaulter" or "non-defaulter." The correct and accurate assignment of these labels is critical for interpreting the data accurately.

Supervised Learning Through Labeled Data

Learning algorithms rely on labeled datasets to discern patterns or relationships between the labels and other dataset attributes. For example, an algorithm may analyze various characteristics (data features) to differentiate between the labels of “defaulter” and “non-defaulter.” In essence, the label directs the learning algorithm, guiding or supervising it to “comprehend” the associations (if any) between the labels and the provided features. The use of labels as a guiding mechanism in learning algorithms is precisely why we refer to these methods as supervised learning algorithms: the label effectively supervises and directs the learning process. The accuracy of these labels is critical to the efficacy of the learning algorithm. Precise and consistent labeling is necessary for the algorithm to discern correct patterns and make reliable predictions.

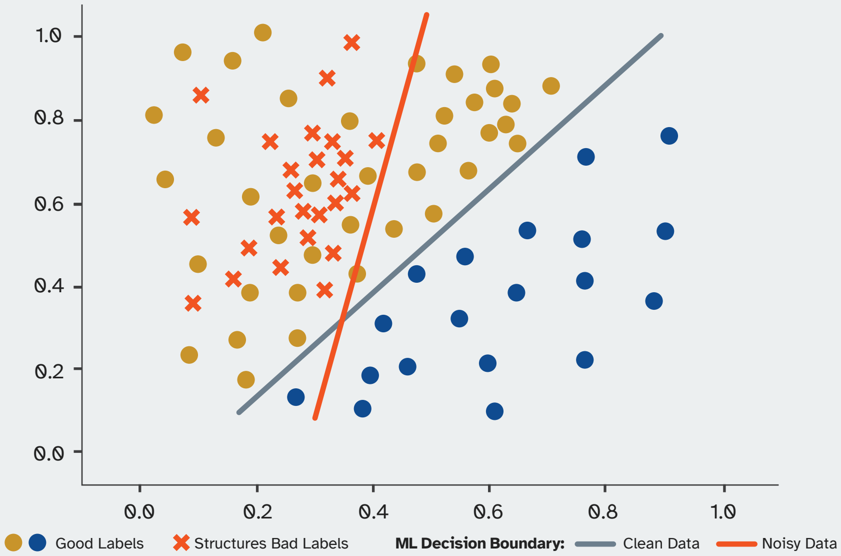
Incorrect Labels/Noisy Labels

While labeled datasets are foundational to many learning algorithms, the reliability of these labels is often a concern. Inaccuracies in labeling can arise from various factors. While this study does not delve into all possible causes of incorrect labeling, it is worth noting that specific issues, such as gender biases or erroneous decision-making during credit evaluation, can lead to the presence of noisy labels in a dataset. Instances where creditworthy applicants are unjustly rejected are particularly relevant to our study and contribute to the noisiness of the dataset.

NOTE

In this study, the terms "incorrect labels" and "noisy labels" are used interchangeably, as they both refer to the same concept.

FIGURE 1. COMPARATIVE ANALYSIS OF ML MODELS WITH OR WITHOUT NOISE INFLUENCE



ML models with noisy labels may not learn the correct decision boundary.

Image Source: Author's rendition of iMerit, March 2021.

Identification of Incorrect Labels

Numerous methods have been developed to identify incorrectly labeled records or observations in datasets. The research by Ghosh et al. (2017), Malach and Shalev-Shwartz (2017), Huang et al. (2019), and Pleiss et al. (2020) stands out in this area. However, readers may question the applicability of these methods, which were initially designed for detecting labeling inaccuracies, to the specific context of reject inference. The following section clarifies the connection between reject inference and incorrect label detection. Following this foundation, the paper turns to introduce counterfactual learning, and the Noisy Label Detection and Counterfactual Correction (NDCC) algorithm developed by Qi, Wenting and Charalampos Chelmis (2023). At the time when this paper is being written and published, NDCC is considered a leading-edge method for incorrect label correction, as opposed to mere detection. This report will further discuss how NDCC can be effectively used to tackle the challenges of reject inference.

Connection Between Reject Inference and Noisy Label Detection

Fundamentally, reject inference utilizes quantitative methods to pinpoint applicants who, despite having been rejected, could in fact be creditworthy. These are individuals who ideally should have been classified as favorable applicants and granted the “approved” label. Situations like these represent instances of detecting incorrect labels. The algorithm I introduce in this section leverages counterfactual correction to identify such noisy labels. Since reject inference is primarily about identifying inaccurately labeled data, this methodology is especially well-suited for tackling the complexities inherent in reject inference.

> In essence, the reject inference challenge is to identify applicants who have been assigned incorrect labels.

Fundamentals of Counterfactual Learning

Counterfactual learning is an ML method employed to create explanations of automated decisions in a way that is understandable to humans. A prime application of this technique is in the domain of credit assessment, where it enables FSPs to offer clear explanations to applicants regarding which specific attributes significantly influenced their credit decision. These insights can offer applicants actionable advice on how to improve their chances for a favorable decision in future credit applications. Such feedback is advantageous for both the FSP and the applicant, fostering transparency and understanding in the credit assessment process.

FIGURE 2. CREDIT DECISION MAKING BASED ON A CUSTOMER'S PROFILE

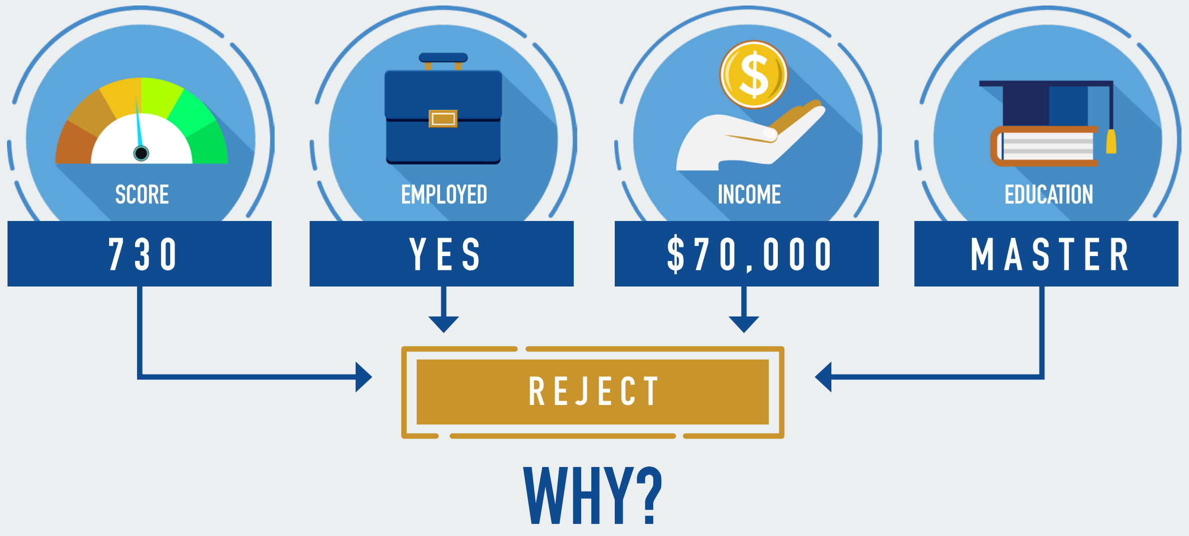


Image Source: Author rendition of IDIAS Lab at SUNY Albany, January 2024.

To illustrate the application of counterfactual learning in credit assessment, let us consider a straightforward example. At its core, counterfactual learning is a task built based on the assumption that the ML model remains unchanged over time. Under this assumption, counterfactual analysis can reliably predict the outcome of future applications from an applicant that changes her financial behavior according to the counterfactuals provided to her at the time her application was rejected. In Figure 3, we observe how the ML model establishes an optimal decision boundary to differentiate between loans that are approved and those that are rejected.

FIGURE 3. MAKING CREDIT DECISIONS BASED ON ML MODEL'S CLASSIFICATION BOUNDARIES

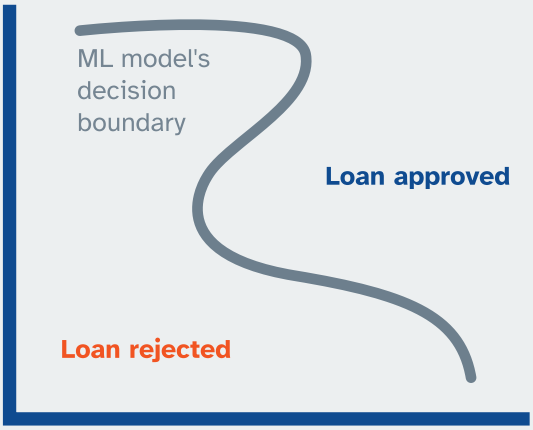


Image Source: Author rendition of IDIAS Lab at SUNY Albany, January 2024.

The pivotal question in using counterfactual learning for credit assessment is identifying which attributes of an applicant need to change for them to cross from the rejection zone into the approval zone within the ML model's decision boundary. This transition is depicted in Figure 4 and Figure 5 as moving from the left side (rejection region) to the right side (approval region). To address this, the counterfactual model searches for datapoints near the one being analyzed, providing explanations in one of two forms: datapoints with the same prediction as the original datapoint, or those with different outcomes. These findings could lead to actionable insights, framed as hypothetical scenarios, such as what might happen if an applicant had obtained a higher degree, or had an annual income of $40,000 instead of $30,000.

For example, Figure 4 demonstrates that an applicant (represented as a blue datapoint) would achieve loan approval by boosting their annual income by $10,000, while Figure 5 outlines a scenario where a previously rejected applicant could secure loan approval by increasing their annual income by $5,000, and at the same time, extending their credit history by one year.

Counterfactual learning shows that a rejected credit applicant has various potential paths to transition across the decision boundary into the acceptance zone. A crucial aspect of counterfactual analysis is identifying minimal changes that can pivot the outcome towards approval. For instance, suggesting a modest income increase of $1,000 for loan approval is far more feasible than demanding a $10,000 hike, or requiring homeownership over renting. Additionally, focusing on altering a single attribute rather than recommending multiple changes, makes the advice more practical and achievable.

FIGURE 4. THE PATH TO APPROVAL WITH A $10,000 INCREASE IN ANNUAL INCOME

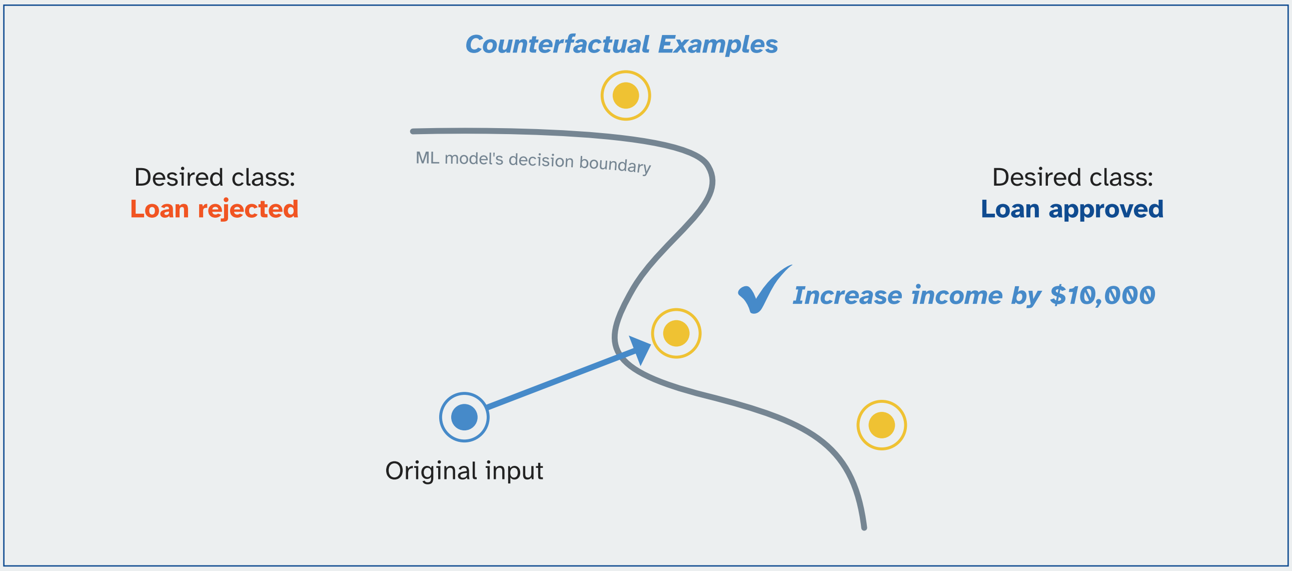


Image Source: Author rendition of IDIAS Lab at SUNY Albany, January 2024.

FIGURE 5. THE PATH OF APPROVAL: A $5,000 INCOME INCREASE AND ONE MORE YEAR OF CREDIT HISTORY

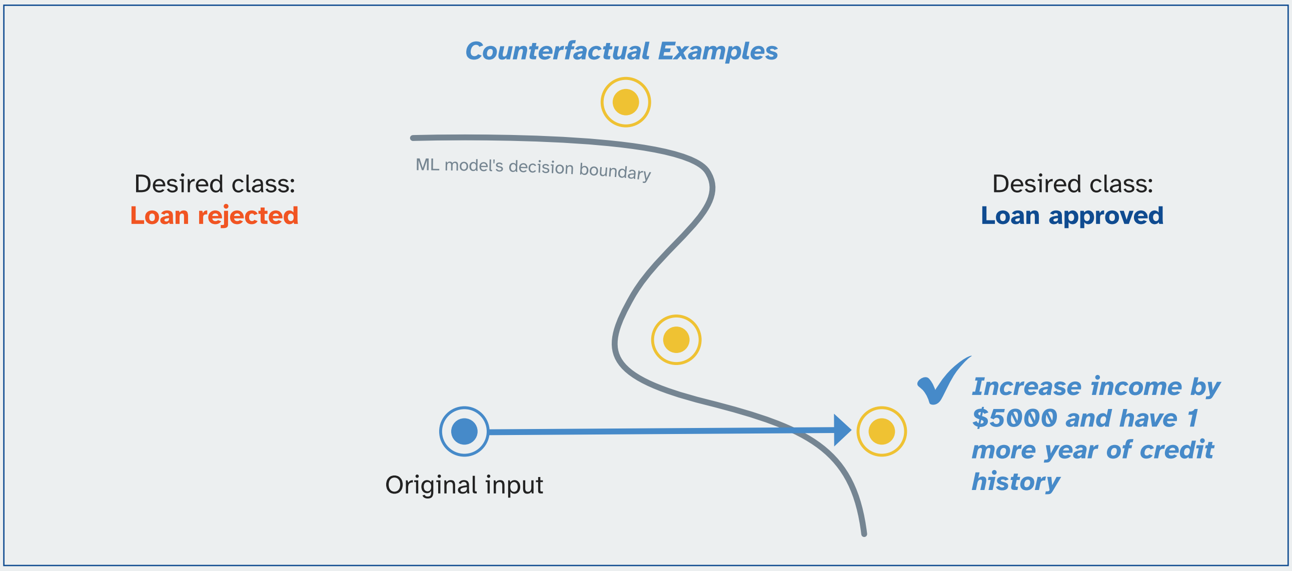

Counterfactual Examples
ML model's decision boundary
Desired class:
Loan rejected
Desired class:
Loan approved
Increase income by $5000 and have 1 more year of credit history
Original input

Image Source: Author rendition of IDIAS Lab at SUNY Albany, January 2024.

Figure 4 and Figure 5 show how counterfactual learning can offer guidance for a rejected applicant to transition from the rejection region into the approval area.

Figure 6 demonstrates how counterfactual learning employs an optimization equation that incorporates two types of distance measurements: one between the model's prediction for the counterfactual and the desired outcome, and the other between individual data points and their respective counterfactuals. In addition, counterfactual explanations need to be measurable and, ideally, actionable. These principles are typically incorporated into the model as constraints within a minimization problem, as illustrated in Figure 6.

Having introduced the foundational concepts of supervised learning and counterfactual learning, I will now delve into the noisy label detection algorithm mentioned earlier in this section. This discussion will explain how the algorithm leverages counterfactual learning to address the reject inference problem.

FIGURE 6. THE OPTIMIZATION EQUATIONS UNDERNEATH THE COUNTERFACTUAL MODEL

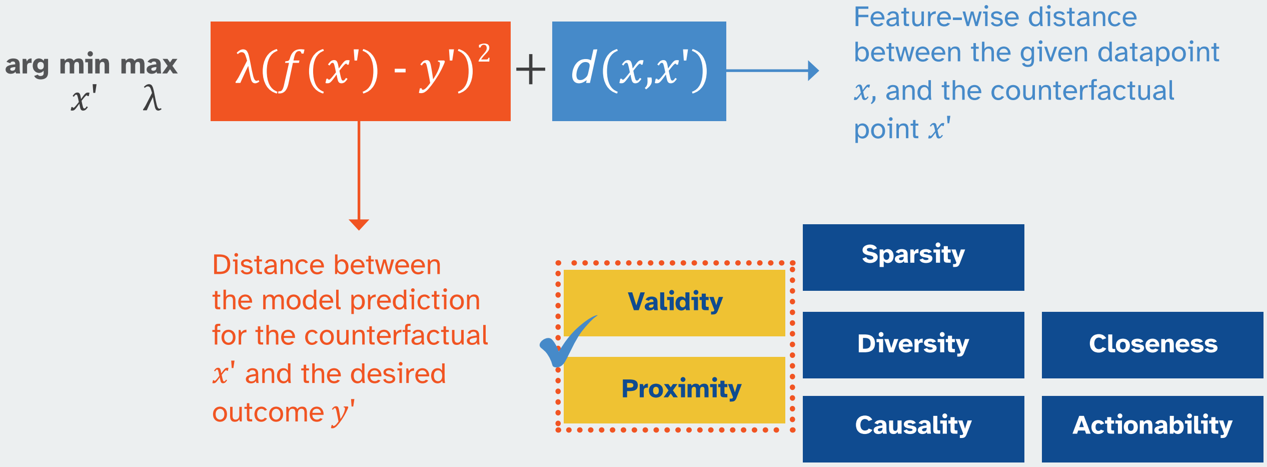

$\arg\min_{x'}\max_{\lambda}$
$\lambda(f(x') - y')^2 + d(x,x')$
Feature-wise distance between the given datapoint x, and the counterfactual point x'
Distance between the model prediction for the counterfactual x' and the desired outcome y'
Validity
Proximity
Sparsity
Diversity
Closeness
Causality
Actionability

Image Source: Author rendition of IDIAS Lab at SUNY Albany, January 2024.

Noisy Label Detection and Counterfactual Correction (NDCC)

Before delving into the noisy label detection algorithm and its application in reject inference, it is essential to familiarize yourself with some key terms and concepts that will be frequently referenced throughout this discussion. I will start with a summary of the concepts used in this approach, followed by an introduction to the main method.

NDCC – Introduction

This section will delve into the NDCC algorithm. Note that this discussion on adapting NDCC to the reject inference problem will primarily focus on its key concepts and methodologies. The goal here is to present this information in a manner that is both clear and approachable, steering clear of overly complex technical details.

FIGURE 7. NOISY LABEL DETECTION AND COUNTERFACTUAL CORRECTION (NDCC) TRAINS A MODEL BY DETECTING AND CORRECTING THE NOISY LABELED DATA POINTS

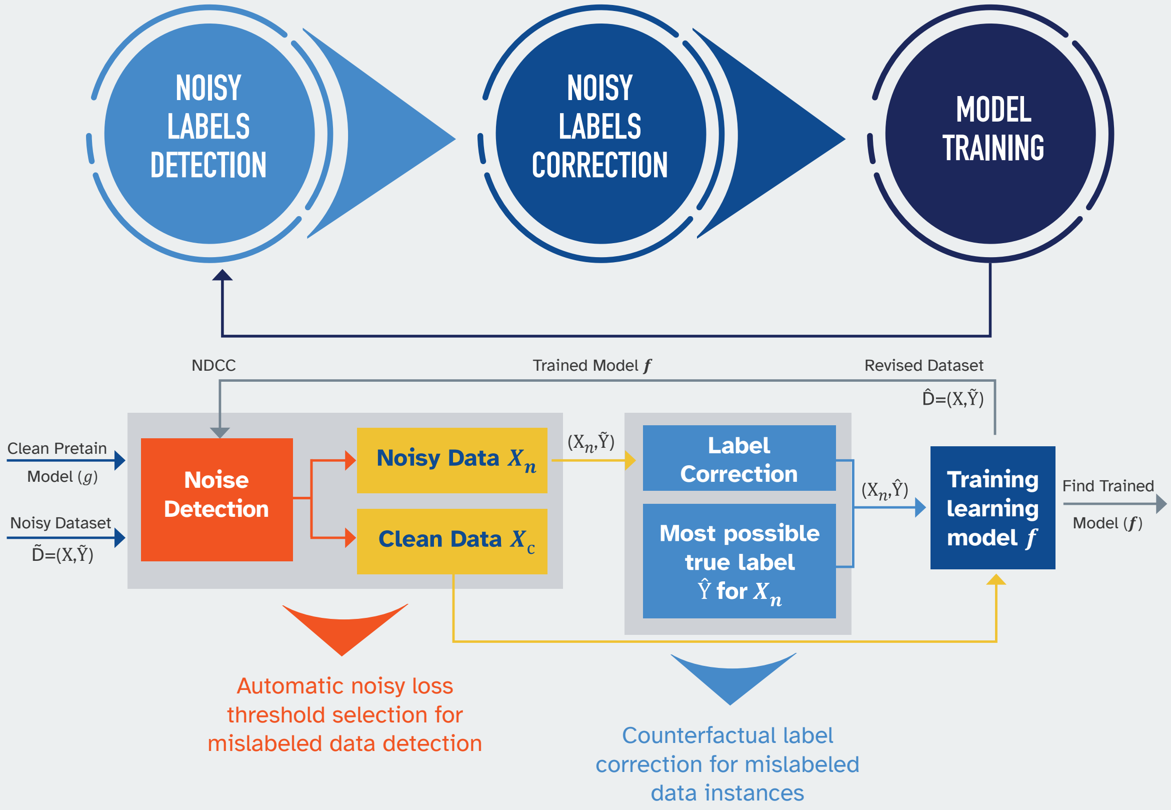

NOISY LABELS DETECTION
NOISY LABELS CORRECTION
MODEL TRAINING
NDCC
Trained Model f
Revised Dataset
$\hat{D}=(X,\hat{Y})$
Clean Pretain Model (g)
Noisy Dataset
$\tilde{D}=(X,\tilde{Y})$
Noise Detection
Noisy Data X_n
Clean Data X_c
$(X_n,\tilde{Y})$
Label Correction
Most possible true label $\hat{Y}$ for X_n
$(X_n,\hat{Y})$
Training learning model f
Find Trained Model (f)
Automatic noisy loss threshold selection for mislabeled data detection
Counterfactual label correction for mislabeled data instances

Image Source: Author rendition of IDIAS Lab at SUNY Albany, January 2024.

NDCC – Objective and Main Steps

The two goals of NDCC are to use counterfactual learning to create a robust model that accurately detects noisy labels (erroneous rejections) and corrects such labels so a more accurate prediction model can be learned. NDCC utilizes an iterative approach, meaning it does not pinpoint all noisy labels in just one go. Rather, it progresses through a series of iterations, the details of which will be explained shortly. The model continues this iterative process until it meets predetermined criteria. Once these criteria are satisfied, the process ends, yielding a dataset where the labels have been refined and corrected. The algorithm comprises three primary steps:

- ✓ **Identifying potentially noisy labeled data instances**: First, the method looks for any data points in the training dataset and that might have incorrect/noisy labels.
- ✓ **Finding the truth**: For each of these potentially mislabeled data points (credit applicants), NDCC guesses what their correct label should be. This step involves generating what-if scenarios and imagining different outcomes in order to better estimate the true label.
- ✓ **Finalizing the model and data**: Lastly, the method offers two outputs: a model trained to classify credit applications as approved or rejected, and an updated dataset in which the labels that were likely wrong have been fixed.

NDCC – Notation and Terminology

Before delving into NDCC and its application to reject inference, I will clarify some key terms that will be used frequently in this discussion:

- **Clean dataset**: This refers to a dataset in which all labels are accurate. In the context of this study, this means a dataset where every rejected applicant has been correctly denied. If these individuals had received a loan, they would have likely defaulted. We use $D(X, y)$ to denote this dataset, where X represents the feature vector with M dimensions—encompassing various factors like income, and credit score—that are considered in the credit evaluation process. We assume that there are M distinct features in assessing each applicant's creditworthiness. The creditworthiness of an applicant is represented by a binary variable, denoted as 'y', which can take on one of two values: approved or rejected.
- **Noisy dataset**: A noisy dataset contains some cases that were incorrectly rejected (or incorrectly approved). In other words, if these applicants had been granted a loan, some of them would most likely have been non-defaulters. Similarly, incorrectly approved applicants would most likely have been defaulters. In this analysis, we treat cases that were mistakenly rejected, often due to errors or biases, as "noise" in the dataset. This dataset is shown with $\tilde{D}(X,\tilde{Y})$.
- **N:** An important term to understand in this study is 'N,' which represents the total number of data instances (credit applications) in $\tilde{D}$. In the context of this work, it is assumed that $\tilde{D}$ is obtained from an ideal clean dataset D of the same size. The goal is thus to retrieve D from $\tilde{D}$.
- **K:** The total number of classes in both Y and Y' are K. In our case K = 2 (approved vs. rejected applicants).
- **f:** The symbol 'f' denotes the model we develop using training (potentially noisy) data, in order to predict the true label 'y' for new, unobserved data instances in the test dataset. Essentially, our objective is to train a model, represented by 'f', to accurately determine the true status (whether rejected erroneously or correctly) of each applicant. This is achieved using financial institutions' available historical data about their approved and rejected applicants, which we refer to as noisy data.
- **g:** The symbol 'g' denotes the model we use to detect noisy labeled data. This model uses the same architecture as f, but is pre-trained using a small subset of high fidelity data obtained from $\tilde{D}$. Specifically, we recommend 10% of applications in $\tilde{D}$ to be carefully inspected for label inconsistences. Identified inconsistences must be fixed, and a clean dataset D_{pre} is to be used to train model g.
- **Y-hat ($\hat{Y}$) :** This represents the predicted outcome generated by our model, denoted as 'f.' In our specific context, Y-hat is a binary variable that takes one of two possible values (approve or reject), and indicates the predictions made for each applicant.

NDCC – Setting Up the Algorithm for Tackling the Reject Inference Problem

NDCC uses a dataset containing both correctly and incorrectly rejected credit applicants, and the use of this dataset develops a robust algorithm (classifier denoted as f) that enables us to detect the credit applicants who were unfairly rejected. To accomplish this goal, NDCC breaks down the main problem, which is finding the noisy labels into smaller sub-problems. Below, is a conceptual overview of these sub-problems—focusing more on the concepts rather than on the intricate technical details—followed by a more detailed exploration.

NDCC Sub-Problems

Initially, the noisy dataset ($\tilde{D}(X,\underset{\sim}{\tilde{Y}})$), containing the data of credit applicants, is fed into the NDCC (Figure 7). The NDCC employs a loss function to assess the loss for each data point, covering both noisy and non-noisy labels. This function acts as a measure of each data point's deviation from its true label. Following this, the NDCC features a module known as the label counterfactual correction module. This module is responsible for assigning the label that is most likely to be accurate based on the features of each applicant and the current model f, essentially predicting the applicant's true label or creditworthiness. After determining the most likely label, the algorithm updates the original label to this more accurate one. Label correction within NDCC is an ongoing process, with potential updates occurring throughout the training phase as additional instances of noisy labels are uncovered. This iterative procedure continues until it reaches a specific stopping criterion.

With this overview in mind, we can now begin to address each sub-problem in detail.

Sub-Problem 1

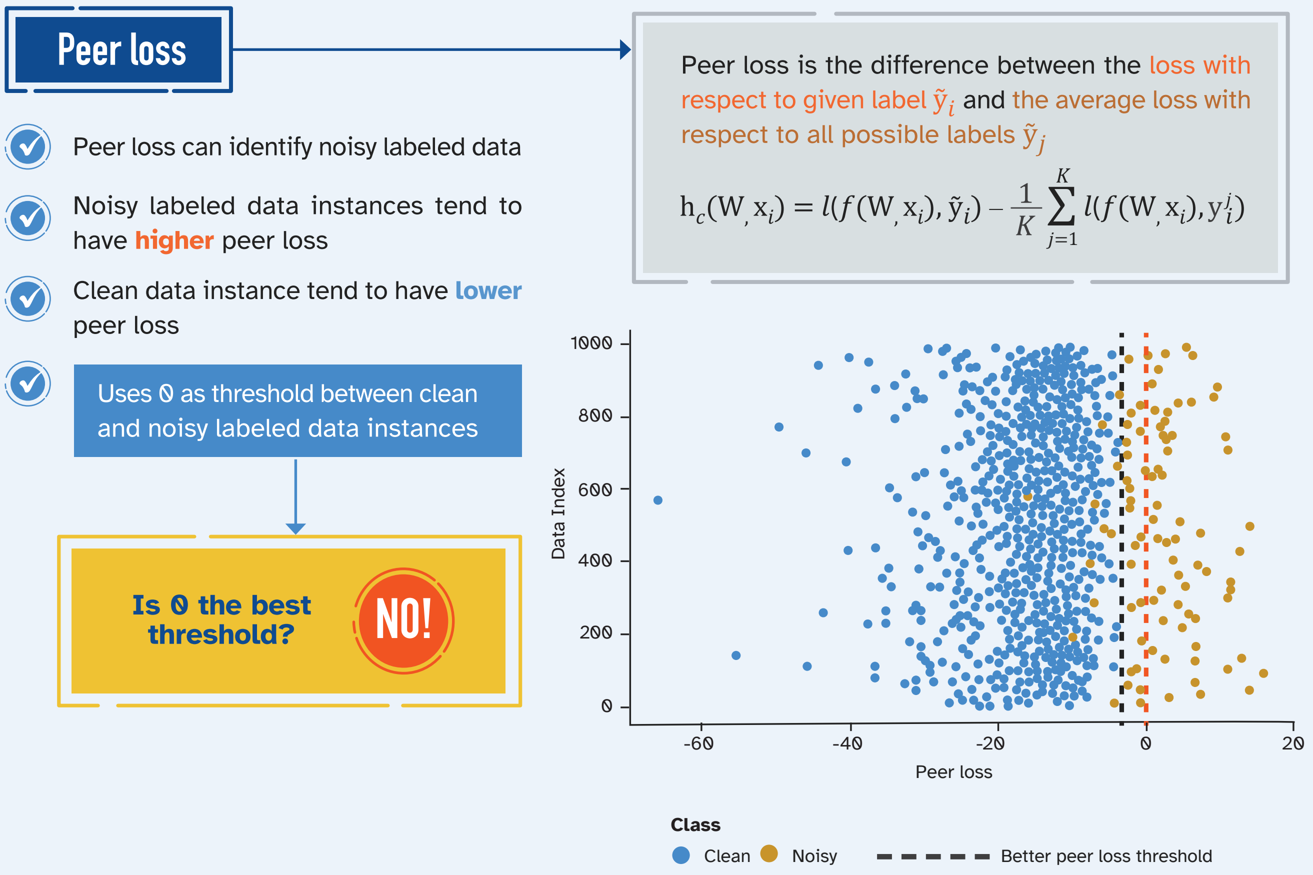
Noisy Label Detection

The main method for detecting noisy labels within a dataset is through the use of the loss function. The loss function serves as a crucial metric for identifying noisy labels, effectively evaluating each applicant's data point to produce values that range from low to high. This facilitates the identification of inaccuracies in label assignment. Generally, the output of the loss function (loss values) for correct credit decisions (both approvals and rejections) tends to be lower than that of incorrect credit decisions. This is typically because noisy labeled data can often represent outliers when compared to the distribution of clean data.

Having introduced the groundwork of loss function, as well as its role in identifying potentially erroneous credit decisions, our focus now shifts to discerning when a loss value should be classified as either small or large. This distinction is key: As noted above, the decision about whether an applicant has been the subject of an incorrect evaluation hinges on the magnitude of their loss value. Making such a decision necessitates a solid and reliable method, known as the “loss threshold.” Using a loss threshold helps us distinguish between credit applicants who were correctly rejected and those who were mistakenly rejected. This approach is grounded in the principle that we need to analyze the dataset of credit applicants to learn the overall distribution of loss values. The loss threshold will differ across the datasets available from various financial institutions. Moreover, NDCC includes an adaptive and intelligent thresholding technique, which implies that the threshold value will not remain constant, but will adjust as required by the data.

The loss threshold employs a specific loss function known as peer loss (Equation 1). Utilizing this peer loss, an objective function is used that aids in determining the appropriate threshold. Figure 8 provides an illustration of Sub-Problem 1, focusing on noisy label detection.

FIGURE 8. THE FIRST PROBLEM TO SOLVE UNDER NDCC IS TO DETECT NOISY LABELS THROUGH PEER LOSS



Peer loss

- ✓ Peer loss can identify noisy labeled data
- ✓ Noisy labeled data instances tend to have **higher** peer loss
- ✓ Clean data instance tend to have **lower** peer loss
- ✓ Uses 0 as threshold between clean and noisy labeled data instances

Is 0 the best threshold? NO!

Peer loss is the difference between the loss with respect to given label $\tilde{y}_i$ and the average loss with respect to all possible labels $\tilde{y}_j$

$$h_c(W, x_i) = l(f(W, x_i), \tilde{y}_i) - \frac{1}{K}\sum_{j=1}^{K} l(f(W, x_i), y_i^j)$$

Image Source: Author rendition of IDIAS Lab at SUNY Albany, January 2024.

Sub-Problem 2

Noisy Label Threshold Selection Criterion

NDCC is sensitive to initial conditions, requiring a strategically selected starting point to steer it towards a robust solution. Choosing the right initial data enhances the model's efficiency in correctly identifying rejection errors. This process involves integrating a subset of records known for their clean and noise-free labels. We denote this subset of cleanly labeled data as D-pre.

We employ D-pre to pre-train a model, which we have named 'g'. Clean data instances in D-pre are derived from a small subset of D̃, which is meticulously examined for label correctness. D-pre should comprise records of individuals who both received loans as well as individuals whose applications were declined. This initial phase of training using D-pre lays a solid foundation for both identifying noisy labels and for learning an accurate model f, ensuring robust analysis and accurate predictions in subsequent stages.

This pre-trained model, 'g', is crucial for learning the parameters used for detecting mistakenly rejected applicants. These parameters act as guidelines, steering our model's learning process in the right direction and narrowing down the universe of possible solutions. By doing so, we limit our search to areas where the correct solution or model parameters are likely to be found. Using D-pre, we apply our peer loss function to all the credit application datasets that are in this clean set. This step would give us the loss for each data instance within the clean dataset. Let us call this the loss-pre-correct. The benefit of these loss values is that they can guide us to make an association between the magnitude of each loss and the credit applicant's true label.

NOTE

Using D-pre to construct a model is essentially the same strategy we referred to earlier: the need for a good initial condition to establish a starting point for the algorithm.

NOTE

The true labels of credit applicants within D-pre are known and verified.

To derive additional usage from these losses gained from using D-pre, we randomly select 10% of the D-pre and then add noise to that percentage of randomly selected records, meaning that we switch the labels of those records. If the label is rejected, we switch it to approved, and if the label is approved and they were good customers, we switch it to rejected. After introducing this noise, we have a dataset in which 10% of the labels are noisy and the remaining labels are correct. Again, we calculate the loss using our loss function for this sample, and we call it loss-pre-noisy. We calculate the difference between these two losses (before and after adding the noise) and call it loss-diff.

Consider that it is expected for the absolute value of the loss difference for noisy data instances to be higher than for the clean data instances. We do need to consider that in subsequent steps of our model training (i.e. without using D-pre), we have no prior indication about which data instances are clean or noisy.

However, the developers of NDCC have shown that there are some ranges in the potential values that loss-diff can take, in which it is more likely to see the noisy labels. This point is one of the main factors that NDCC uses to optimize its decision-making threshold for distinguishing erroneous credit rejections.

A demonstrated in Figure 8, Sub-Problem 2 involves the following steps:

1. Identify the peer loss area in which mislabeled data instances are more likely to be found.
2. Iteratively computes the peer loss value distribution using a pretrained model using D-pre.
3. Determines the threshold by averaging their peer loss value:

$$thr = \frac{1}{|\tilde{D}_{ns}|} \sum_{i} h_c(W_{g,} x_i)_, x_i \in \tilde{D}_{ns}$$

Equation 1. Peer loss

As a result of these three steps, it is expected that mislabeled data instances congregate in the same area (red area shown in Figure 9).

FIGURE 9. AUTOMATED THRESHOLD SELECTION ACROSS TRAINING ITERATIONS REDUCES LABEL NOISE IN DATA

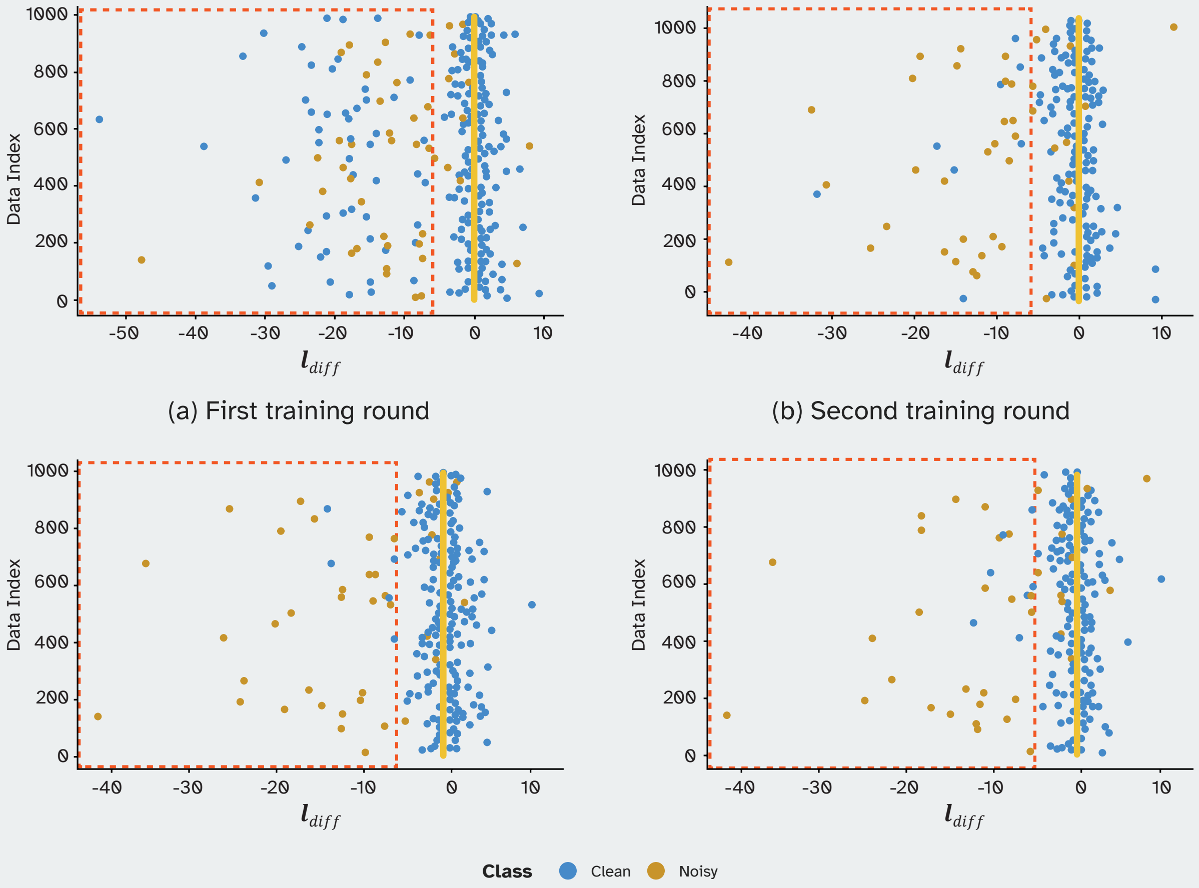


Image Source: Author rendition of IDIAS Lab at SUNY Albany, January 2024.

Sub-Problem 3

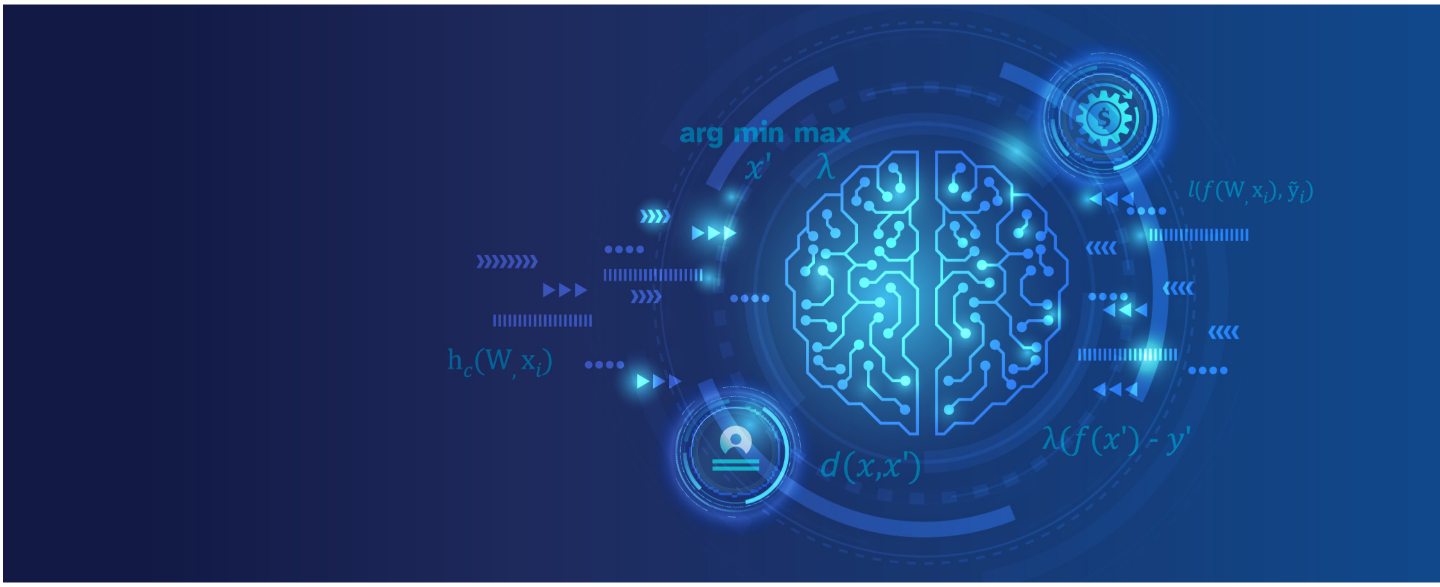
Noisy Label Correction

The noisy label correction process is designed to pair suspected rejected applicants (the cases in which applicants may have been mistakenly rejected) with their most likely true label using counterfactual learning. NDCC generates a counterfactual data instance with other possible labels for each detected case of noisy labeled data. This correction provides the opportunity to calculate the loss for each data instance. Using this loss, NDCC decides on the label considered the most likely to be correct—in other words, the label with the smallest loss value -- and if necessary, correct the label that is associated with that data point.

Applying this methodology to the reject inference problem enables the use of loss values to determine the likelihood of a rejection being accurate. Furthermore, the counterfactual capabilities of NDCC allow for exploration into how modifications in a customer's profile could influence a shift from rejection to approval. This step also facilitates the examination of potential biases against protected attributes. For instance, altering the gender or race of a rejected applicant in a counterfactual analysis—if it results in an approval—signals the existence of bias.

Conclusions

Addressing reject inference bias can help institutions mitigate gender biases and prevent the erroneous rejection of creditworthy applicants, leading to both ethical and business benefits.



This paper has focused on exploring reject inference. Throughout this research, I prioritized the practical application of the methods, aiming to encourage financial institutions to adopt these techniques. Addressing reject inference bias can help institutions mitigate gender biases and prevent the erroneous rejection of creditworthy applicants, leading to both ethical and business benefits.

The paper introduced two distinct categories of algorithms suitable for the reject inference problem. The first category includes matching algorithms. We discussed how to select the most appropriate algorithm, along with the best matching strategy, to identify erroneously rejected credit applicants. This category of algorithms is intuitive and effective in pinpointing creditworthy applicants.

The second approach, NDCC, is a state-of-the-art AI model designed for noisy label detection. Its application in reject inference is particularly innovative. NDCC is a powerful method capable of yielding robust results using counterfactual learning. While it requires a more advanced understanding of machine learning compared to the first set of algorithms, its implementation may be more complex and time-consuming. NDCC plays a crucial role in identifying incorrect rejections, elucidating the reasons behind them, and uncovering any biases in credit decision-making. It provides valuable insights for financial institutions and applicants alike, outlining the causes of rejections and offering actionable advice for securing future credit approvals.

Whatever method a data scientist chooses to address reject inference bias, the prospective impact of work in this area is tremendous. With the tools outlined in this report, data scientists have the power to enable more women to access credit, opening up opportunity for business growth and community prosperity. Solving reject inference bias is a key ingredient to women's financial inclusion and economic empowerment.

References

Banasik, J., & Crook, J. (2007). Reject inference, augmentation, and sample selection. *European Journal of Operational Research, 183*(3), 1582-1594.

Berg, T., Burg, V., Gombović, A., & Puri, M. (2020). On the rise of fintechs: Credit scoring using digital footprints. *The Review of Financial Studies*, 33(7), 2845-2897.

Celis, L. E., Keswani, V., & Vishnoi, N. (2020, November). Data preprocessing to mitigate bias: A maximum entropy-based approach. *Proceedings of the International Conference on Machine Learning, 119*, 1349-1359.

Ghosh, A., Kumar, H., & Sastry, P. S. (2017, February). Robust loss functions under label noise for deep neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence, 31*(1).

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: a review. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 160(3), 523-541.

Hand, D. J. (2001). Modelling consumer credit risk. *IMA Journal of Management Mathematics*, 12(2), 139-155.

Huang, J., Qu, L., Jia, R., & Zhao, B. (2019). O2u-net: A simple noisy label detection approach for deep neural networks. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3326-3334.

IDIAS Lab. (2024). *Automated threshold selection across training iterations reduces label noise in data*. [Digital image]. PowerPoint presentation.

IDIAS Lab. (2024). *Making credit decisions based on ML model's classification boundaries* [Digital image]. PowerPoint presentation.

IDIAS Lab. (2024). *Noisy label detection and counterfactual correction (NDCC) trains a model by detecting and correcting the noisy labeled data points* [Digital image]. PowerPoint presentation.

IDIAS Lab. (2024). *The path of approval: a $5,000 income increase and one more year of credit history* [Digital image]. PowerPoint presentation.

IDIAS Lab. (2024). *The path to approval with a $10,000 increase in annual income* [Digital image]. PowerPoint presentation.

iMerit. (2021). *How noisy labels impact machine learning models* [Digital image]. https://imerit.net/blog/how-noisy-labels-impact-machine-learning-models/

Iosifidis, V., Fetahu, B., & Ntoutsi, E. (2019, December). Fae: A fairness-aware ensemble framework. *IEEE International Conference on Big Data*, 1375-1380.

Kelly, S., & Mirpourian, M. (2021). Algorithmic bias, financial inclusion, and gender: A primer on opening up new credit to women in emerging economies. Women's World Banking.

Li, Z., Tian, Y., Li, K., Zhou, F., & Yang, W. (2017). Reject inference in credit scoring using semi-supervised support vector machines. *Expert Systems with Applications, 74*, 105-114.

Iosifidis, V., Fetahu, B., & Ntoutsi, E. (2019, December). Fae: A fairness-aware ensemble framework. International Conference on Big Data (Big Data) (pp. 1375-1380). IEEE.

Malach, E., & Shalev-Shwartz, S. (2017). Decoupling "when to update" from "how to update". *Advances in Neural Information Processing Systems*, 30.

Mancisidor, R. A., Kampffmeyer, M., Aas, K., & Jenssen, R. (2020). Deep generative models for reject inference in credit scoring. *Knowledge-Based Systems*, 196, 105758.

Mirpourian, M., Fu, J., & Kelly, S. (2023). Check your bias! A field guide for lenders. Women's World Banking.

Mitra, G., Hoang, K. T., Gladilin, A., Chu, Y., Black, K., & Mani, G. (2023). Alternative data: Overview. *Handbook of Alternative Data in Finance, 1*, 1-28.

Pleiss, G., Zhang, T., Elenberg, E., & Weinberger, K. Q. (2020). Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems, 33*, 17044-17056.

Qi, W., & Chelmis, C. (2023). Noisy Label Detection and Counterfactual Correction. *Transactions on Artificial Intelligence*. IEEE.

Shelke, M. S., Deshmukh, P. R., & Shandilya, V. K. (2017). A review on imbalanced data handling using undersampling and oversampling techniques. *International Journal of Recent Trends in Engineering and Research, 3*(4), 444-449.

Shen, F., Zhao, X., & Kou, G. (2020). Three-stage reject inference learning framework for credit scoring using unsupervised transfer learning and three-way decision theory. *Decision Support Systems, 137*, 113366.

Zhang, B. H., Lemoine, B., & Mitchell, M. (2018, December). Mitigating unwanted biases with adversarial learning. *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, 335-340.

Women’s World Banking

[Women's World Banking](#) [@womensworldbnkg](#)

[Women's World Banking](#) [@womensworldbnkg](#)

www.womensworldbanking.org