Computational Statistics
https://doi.org/10.1007/s00180-022-01220-9

ORIGINAL PAPER

Semi-supervised adapted HMMs for P2P credit scoring systems with reject inference

Monir El Annas[1] · **Badreddine Benyacoub**[1] · **Mohamed Ouzineb**[1]

Received: 13 May 2021 / Accepted: 20 March 2022

Abstract

The majority of current credit-scoring models, used for loan approval processing, are generally built on the basis of the information from the accepted credit applicants whose ability to repay the loan is known. This situation generates what is called the selection bias, presented by a sample that is not representative of the population of applicants, since rejected applications are excluded. Thus, the impact on the eligibility of those models from a statistical and economic point of view. Especially for the models used in the peer-to-peer lending platforms, since their rejection rate is extremely high. The method of inferring rejected applicants information in the process of construction of the credit scoring models is known as reject inference. This study proposes a semi-supervised learning framework based on hidden Markov models (SSHMM), as a novel method of reject inference. Real data from the Lending Club platform, the most used online lending marketplace in the United States as well as the rest of the world, is used to experiment the effectiveness of our method over existing approaches. The results of this study clearly illustrate the proposed method's superiority, stability, and adaptability.

Keywords Reject Inference · P2P lending · Credit scoring · Hidden Markov models · Semi-supervised learning

✉ Monir El Annas
elannas.mounir@gmail.com

Badreddine Benyacoub
benyacoubb@gmail.com

Mohamed Ouzineb
ouzineb.insea@gmail.com

[1] Institut National de Statistique et d'Economie Appliquée, Rabat, Morocco

Published online: 14 May 2022

1 Introduction

Fintech is emerging rapidly worldwide. Despite the economic shock from the COVID-19 pandemic, global Fin-tech investments remained strong, with over $ 25.6 billion in the first half of 2020 (https://home.kpmg/xx/en/home/insights/2020/02/pulse-of-fintech-archive.html). The pandemic has significantly accelerated digital trends and the demand for digital platforms such as digital banking, peer-to-peer lending platforms and other fintech-related services. The peer-to-peer lending (P2P lending) online platforms (https://www.lendingclub.com/info/download-data.action), allows borrowers to obtain loans directly from other people. For lenders, it is an alternative to lend customers without going through banks and credit organizations which are very demanding in terms of guarantees and expensive in terms of bank transaction charges. Despite its many advantages, P2P lending is associated with a high level of risk for lenders. As a result, credit scoring systems are commonly used by P2P lending platforms to evaluate potential borrowers. This is generally done by building models using only data from previous accepted applicants without taking into account the applicants who have been rejected. As a result the credit scoring models are biased (Bücker et al. 2013), as well as statistical and economic consequences (Chen and Astebro 2001; Marshall et al. 2010). Reject inference as a method of inferring the credit worthiness status of the rejected applications, has raised a lot of interest in the P2P lending domain, where rejection rate is extremely high. For example, between June 2007 and December 2018, Lending Club P2P lending platform (https://www.lendingclub.com/info/download-data.action), accepted 2;260;701 loans and rejected 27;648;741 loans. As a result, only 8% of loans are issued by the platform. The majority of reject inference methods uses statistical techniques. However, semi-supervised machine learning algorithms are in growing use in this research topic (see Table 1). This study proposes a semi-supervised hidden Markov model (SSHMM) as a novel method to evaluate the usage of semi-supervised machine learning for reject inference in credit scoring. We compare the performance of the SSHMM model with a set of state-of-the-art semi-supervised machine learning algorithms used for reject inference. In addition, supervised machine learning models are used to evaluate the performance gain of reject inference. Finlay, by sampling the rejected data set to generate several samples with varied rejection rates, we conduct a full-sensitivity study on reject inference. The following is a breakdown of the paper's structure. Section 2 discusses related work on credit scoring and reject inference strategies, followed by Sect. 3's discussion of HMM models and introduction to the proposed SSHMM model. Section 4 summarizes the data, experiments sets up, and discusses the major findings. Finally, we give the primary conclusion as well as some suggestions for further research.

2 Literature review

Credit scoring is used by financial institutions and P2P lending platforms, to assess the credit worthiness of loan applicants, usually embedded in a probabilistic framework $p(y \mid x)$, which describes the likelihood that an applicant will repay his loan $(y = 1)$ or not $(y = 0)$ depending on his characteristics x. As a result, estimating $p(y \mid$

x) is an important part of any credit rating process. Generally , the two types of standard credit scoring models, statistical and machine learning based models (Siddiqi 2017; Lessmann et al. 2015), uses only the information on loan records of accepted applicants. The reject inference process of inferring the good or bad loan performance of rejected applicants in the construction of credit scoring models, have been explored as a missing data problem and categorized into three types (Feelders 1999), based on the modelling of $p(z \mid x, y)$, where z is a binary variable which indicates if the applicant has benefited from a credit (his request has been accepted) or the customer has not benefited from a credit (his request has been refused):

The first missing mechanism is missing completely at random (MCAR), which means $p(z \mid x, y) = p(z)$. In this situation, applicants are approved or denied independently of their loan records or personal information, implying that applicants' good or bad behaviour is independent of applicant characteristics x and class y. It basically means that platforms or financial institutions choose whether or not to accept applicants at random, without considering their characteristics or repayment history. As a result, under the MCAR condition, there is no selection mechanism, and thus no sample bias in the lending process. The way platforms and financial institutions handle loan applications is totally inconsistent with this mechanism. As a result, in credit scoring models, it is always disregarded.

The second mechanism is missing at random (MAR), which means $p(z \mid x, y) = p(z \mid x)$. In this situation, loans request are accepted only on the basis of the values of x and certain arbitrary cut-offs. In credit scoring applications, this is similar to $p(y \mid x, z) = p(y \mid x)$.

The third is missing not at random (MNAR), which states that z can be influenced by missing data y, implying that $p(z \mid x, y) \neq p(z \mid x)$. MNAR is a type of missing data in which the result class is determined not just by x but also by y, which is impacted by some unobserved variables, such as loan officers' manual overrides of the model decision (according to their overall impression of an applicant, based on personal experience or other factors). The majority of online loan investors, in particular, are not expert financial investors, and their selections are frequently influenced by a variety of subjective reasons.

In reject inference, a variety of strategies have been used, which may be divided into statistical methods and machine learning techniques. The most common statistical methods used in early reject inference studies are augmentation and extrapolation (Banasik et al. 2003; Anderson 2007). In augmentation, the weights of accepted loan applications are increased by augmenting them. In extrapolation the credit-scoring model is initially built based solely on accepted applications, then predicts the classes of rejected applications before creating a new credit-scoring model based on both samples. However, according to relevant research, augmentation and extrapolation methods do not increase the performance of credit scoring models in most circumstances when compared to the original credit-scoring model trained with solely accepted loans (Banasik and Crook 2007; Crook and Banasik 2004). Survival analysis techniques (Sohn andShin 2006) are another extensively used approach to reject inference. However they have only been found to be of use if there are a majority of rejected applications (Banasik and Crook 2010).

In contrast, some recent studies on reject inference in a semi-supervised scenario have been undertaken based on: The support vector machine (Maldonado and Paredes 2010; Li et al. 2017; Tian et al. 2018; Kim and Cho 2019), Gradient boosting decision tree (Xia et al. 2018), Lightgbm (Xia 2019), Bayesian networks (Anderson 2019), Deep generative models (Mancisidor et al. 2020), Logistic regression (Kozodoi et al. 2019), and Ensemble learning framework that combines multiple classifiers and clustering algorithms (Liu et al. 2020; Shen et al. 2020; Kang et al. 2021). In comparison to statistical approaches, all of the experiments above proved the superiority of semi-supervised machine learning methods of reject inference. A summary of reject inference research using semi-supervised machine learning approaches is shown in Table 1.

3 Methodology

This section introduces the discrete case of hidden Markov models' mathematical basis and learning algorithms. The proposed SSHMM model is then described.

3.1 Hidden Markov models elements

The transition matrix A, the observation probability matrix B, and the initial probability vector pi are the hidden Markov model parameters, which are represented in a single parameter $\lambda = \{A, B, \pi\}$. The main elements of a hidden Markov model are summarized in Table 2 Baum et al. (1970); Levinson et al. (1983); Li et al. (2000).

3.2 Baum–Welch learning for a single observation sequence

In order to ulistrate the Baum–Welch procedure for estimating the parameter $lambda$ of an HMM that generates a single observation sequence , we define the following probabilities (Baum et al. 1970; Levinson et al. 1983):

- The joint probability function $\alpha_t(i) = P\,(o_1, o_2, \ldots, o_t, s_t = e_i \mid \lambda)$, which can be computed recursively as follows (forward algorithm):
  For i $= 1, 2, \ldots,$ N $\quad \alpha_{t=1}(i) = \pi_i b_i\,(o_1)$
  For $t = 2, 3, \ldots, T$, and for $j = 1, 2, \ldots, N, \alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j\,(o_t)$
  Thus, $P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$
- The conditional probability $\beta_t(i) = P\,(o_{t+1}, o_{t+2}, \ldots, o_T \mid s_t = e_i, \lambda)$, which can be computed recursively as follows (backward algorithm):
  For $i = 1, \ldots, N\ \beta_T(i) = 1$
  For $t = T-1, T-2, \ldots, 1,$ for $i = 1, \ldots, N, \beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j\,(o_{t+1})\,\beta_{t+1}(j)$
  Thus, $P(O \mid \lambda) = \sum_{i=1}^{N} \pi_i b_i\,(o_1)\,\beta_1(i)$
- The probability $\gamma_t(i)$ of being in the state e_i at time t as:

$$\gamma_t(i) = P\,(s_t = e_i \mid O, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{P(O, \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}$$

Table 1 Research overview on reject inference using semi supervided machine learning methods

(Year) Author	Reject inference approach	Classification method
Maldonado and Paredes (2010)	Self-training	SVM
Maldonado and Paredes (2010)	Co-training	SVM
Li et al. (2017)	Semi-supervised SVM	S3VM
Tian et al. (2018)	Extrapolation	KFQS- SVM
Xia (2019)	CPLE	LightGBM
Anderson (2019)	Extrapolation	Bayesian networks
Xia et al. (2018)	Extrapolation	Outlier detection- GBT
Kim and Cho (2019)	Label propagation	Transudative SVM
Mancisidor et al. (2020)	Mixture modeling	Mixture modeling-ANN
Kozodoi et al. (2019)	Shallow self-learning	LR
Liu et al. (2020)	SSL-EC3	LR-KNN-SVM-DT-RF
Shen et al. (2020)	Transfer learning-3WD	LR, ANN, RF, XGBoost
Kang et al. (2021)	Label spreading	LR, SVM, RF, XGBoost, LightGBM, GBDT

Table 2 An overview of the main elements of a Hidden Markov model

Element of HMM	Description
The length of the observation sequence	T
The number of states	N
The number of symbols per state	M
The observation sequence	$O = \{o_t, t = 1, 2, \ldots, T\}$
The hidden state sequence	$S = \{s_t, t = 1, 2, \ldots, T\}$
The possible values of each state	$E = \{e_i, i = 1, 2, \ldots, N\}$
The possible symbols per state	$V = \{v_k, k = 1, 2, \ldots, M\}$
The transition matrix	$A = \left(a_{ij}\right), a_{ij} = P\left(s_t = e_j \mid s_{t-1} = e_i\right), i, j = 1, 2, \ldots, N$
The initial probability vector	$\pi = (\pi_i), \pi_i = P\left(s_1 = e_i\right), i = 1, 2, \ldots, N.$
The observation probability matrix	$B = (b_{ik}), b_{ik} = P\left(o_t = v_k \mid s_t = e_i\right), i = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, M.$

- The probability $\xi_t(i, j)$ of being in the state e_i at time t and in the state e_j at time $t+1$,

$$\xi_t(i, j) = P\left(s_t = e_i, s_{t+1} = e_j \mid O, \lambda\right) = \frac{\alpha_t(i) a_{ij} b_j\left(o_{t+1}\right) \beta_{t+1}(j)}{P(O, \lambda)}$$

Thus,

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j)$$

Then, HMM model learning using the Baum–Welch algorithm is done as follows:

The Baum–Welch algorithm for a single observation sequence

1: Initialization: λ parameters, δ tolerance, $\triangle$ gain
2: Repeat until $\triangle < \delta$
- Compute $P(O, \lambda)$
- Compute the new parameters λ^* : for $1 \leq i \leq N$

$$\pi_i^* = \gamma_1(i)$$
$$a_{ij}^* = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, 1 \leq j \leq N$$
$$b_{ik}^* = \frac{\sum_{t=1, o_t = v_k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}, 1 \leq k \leq M$$

- Compute $\triangle = \left|P\left(O, \lambda^*\right) - P(O, \lambda)\right|$
- Adjust $\lambda = \lambda^*$
3: Return: parameters λ.

3.3 Baum–Welch learning for multiple observation sequences

HMM may be extended to support L independent observable variables with one common hidden sequence. To explain the Baum–Welch learning for L independent observation sequences $\left(O^{(1)}, O^{(2)}, \ldots, O^{(L)}\right)$ with equal length T, we first define the following probabilities:

- The joint probability function $\alpha_t^{(l)}(i) = P\left(o_1^{(l)}, o_2^{(l)}, \ldots, o_t^{(l)}, s_t = e_i \mid \lambda\right)$, $i = 1, 2, \ldots, N$; $t = 1, 2, \ldots, T$ and $l = 1, 2, \ldots, L$.
  Which can be calculated for $l = 1, 2, \ldots, L$, recursively, as follows (forward algorithm) :
  For $i = 1, 2, \ldots, N$ $\alpha_1^{(l)}(i) = \pi_i b_i\left(o_1^{(l)}\right)$
  For $t = 2, 3, \ldots, T$, and for $j = 1, 2, \ldots, N$, $\alpha_t^{(l)}(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j\left(o_t^{(l)}\right)$

Thus, $P\left(O^{(l)} \mid \lambda\right) = \sum_{i=1}^{N} \alpha_T^{(l)}(i)$ and $P(O \mid \lambda) = \prod_{l=1}^{L} P\left(O^{(l)} \mid \lambda\right)$

- The conditional probability $\beta_t^{(l)}(i) = P\left(o_{t+1'}^{(l)} o_{t+2}^{(l)}, \ldots, o_T^{(l)} \mid s_t = e_i, \lambda\right)$
  $i = 1, 2, \ldots, N; t = 1, 2, \ldots, T$ and $l = 1, 2, \ldots, L$.
  Which can be calculated for $l = 1, 2, \ldots, L$, recursively, as follows (backward algorithm):
  For $i = 1, 2, \ldots, N$ $\beta_T^{(l)}(i) = 1$
  For $t = T - 1, T - 2, \ldots, T$, and for $j = 1, 2, \ldots, N$, $\beta_t^{(l)}(i) = \sum_{j=1}^{N} a_{ij} b_j \left(o_{t+1}^{(l)}\right) \beta_{t+1}^{(l)}(j)$
  Thus, $P\left(O^{(l)} \mid \lambda\right) = \sum_{i=1}^{N} \beta_1^{(l)}(i)$ and $P(O \mid \lambda) = \prod_{l=1}^{L} P\left(O^{(l)} \mid \lambda\right)$
- The probability $\gamma_t^{(l)}(i)$ of being in state e_i at time t, given the observation $O^{(l)}, l = 1, 2, \ldots, L$:

$$\gamma_t^{(l)}(i) = P\left(s_t = e_i \mid O^{(l)}, \lambda\right) = \frac{\alpha_t^{(l)}(i)\beta_t^{(l)}(i)}{P\left(O^{(l)} \mid \lambda\right)} = \frac{\alpha_t^{(l)}(i)\beta_t^{(l)}(i)}{\sum_{i=1}^{N} \alpha_t^{(I)}(i)\beta_t^{(l)}(i)}$$

- The probability $\xi_t^{(l)}(i, j)$ of being in state e_i at time t and state e_j at time $t + 1$, given the observation $O^{(l)}, l = 1, 2, \ldots, L$:

$$\xi_t^{(l)}(i, j) = P\left(s_t = e_i, s_{t+1} = e_j \mid O^{(l)}, \lambda\right) = \frac{\alpha_t^{(l)}(i) a_{ij} b_j \left(o_{t+1}^{(l)}\right) \beta_{t+1}^{(l)}(j)}{P\left(O^{(l)}, \lambda\right)}$$

Thus,

$$\gamma_t^{(l)}(i) = \sum_{j=1}^{N} \xi_t^{(l)}(i, j)$$

Then, HMM model learning is done using the Baum–Welch algorithm as follows:

3.4 Semi-supervised HMM adapted for credit scoring with reject inference

We propose a semi-supervised hidden Markov model (SSHMM) framework to address the problem of reject inference, which aims at taking advantage of the data collected on both accepted and rejected credit applicants. The proposed SSHMM model construction is done in three main stages: binning, filtering, and model training.

In the first stage, a binning process is used to discretize the values of continuous variables into bins and address the presence of outliers and statistical noise. Furthermore, the binning process is used for data scaling and model complexity reduction. It is worth noting that binning techniques are commonly applied in credit risk modelling (Siddiqi 2017). The binning quality is assessed using a score, considering the following aspects (Navas-Palencia 2020) : information value (IV), statistical significance and homogeneity.

The Baum–Welch algorithm for multiple observation sequences

1. Initialization: λ parameters, δ tolerance, $\triangle$ gain
2. Repeat until $\triangle < \delta$

- Compute $P(O, \lambda) = \Pi_{l=1}^{L} P\left(O^{(l)} \mid \lambda\right)$.
- Compute new parameters λ^*, for $1 \leq i \leq N$

$$\pi_i^* = \frac{1}{L} \sum_{l=1}^{L} \gamma_1^{(l)}(i)$$

$$a_{ij}^* = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T-1} \xi_t^{(l)}(i, j)}{\sum_{l=1}^{L} \sum_{t=1}^{T-1} \gamma_t^{(l)}(i)}, 1 \leq j \leq N$$

$$b_i(k)^* = \frac{\sum_{l=1}^{L} \sum_{t=1}^{T} \Big|_{o_t^{(l)} = v_k^{(l)}} \gamma_t^{(l)}(i)}{\sum_{l=1}^{L} \sum_{t=1}^{T} \gamma_t^{(l)}(i)}, 1 \leq k \leq M$$

- Compute $\triangle = P\left(O, \lambda^*\right) - P(O, \lambda)$
- Adjust $\lambda = \lambda^*$

3. Return: parameters λ.

In the second stage, a filtering process is performed to remove observations that may have a deleterious effect on the model's performance, using isolation forest algorithm (Liu et al. 2008). We first remove rejected applicants that different the most of the accepts distribution. Second, rejected applicants who are the most identical to those who have been accepted are removed. Furthermore, the filtering process, reduce data noise and retain clean data, thus decrease data size and save computing resources.

In the third stage, the HMM structure is set such that the class labels (good/bad) is represented by two hidden states and the observation sequence corresponds to the sequence of observation resulting from the binned characteristics. We first compute the initial parameter λ of HMM, using maximum likelihood estimation (MLE), as the following counts :

$$a_{ij} = \frac{Number\ of\ transitions\ from\ state\ e_i\ to\ state\ e_j}{Number\ of\ transitions\ out\ state\ e_i}$$

$$b_{ik} = \frac{Number\ of\ times\ in\ state\ e_i\ and\ symbol\ v_k\ occurs}{Number\ of\ times\ in\ state\ e_i}$$

$$\pi_i = \frac{Number\ of\ times\ in\ state\ e_i}{Number\ of\ observations}$$

Then, we adjust HMM parameters using the iterative procedure of Baum–Welch learning given the observed sequences from rejected applicants samples. The flow chart describing the SSHMM modelling pipeline is presented in Fig. 1. Thus, SSHMM take advantage of unsupervised learning and supervised learning. As such, it adds together information from unsupervised learning (using the BWA) and supervised learning (using MLE) to get the complete model. Since the initialization is done in supervised manner, the learned parameters will always be in alignment with the initialization labels instead of randomly assigned labels. As a result, a more consistent credit scoring model with reject inference.

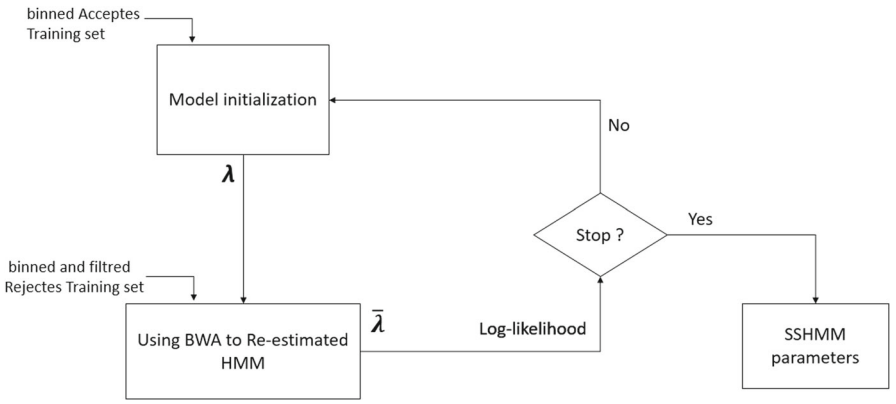


Fig. 1 The flow chart for creating SSHMM model

4 Experimental setup and results

The data sets, performance measures, and the evaluation baseline of the proposed framework are all introduced in this section.

4.1 Data and variables

Our numerical experiment was based on data from Lending Club online credit marketplace (https://www.lendingclub.com/info/download-data.action), for the period from 2007 until 2018 and contain both rejected and accepted applications. Since the characteristics of the accepted and rejected data sets were incompatible. The accepted data set initially had 150 characteristics. However the rejected data set only has six: loan amount, fico score, debt-to-income (dti) ratio, loan purpose, address state, and employment length. Only the aforementioned characteristics shared by accepted and rejected applicants are used in this study. Although the rejected data sets features provide a lot of information about applicants' creditworthiness, if only the six characteristics were used to build the credit scoring model, some important information might have been missed. Only loans with a completely paid or defaulted status were considered, and records with missing values or obvious errors were removed. The final data set used in this study contain 2;064;314 rejected loans and 1;266;782 accepted loans, including 247;426 default loans. Tables 3 and 4 shows descriptive statistics of the Lending Club data. The data binning summary is given in Table 5. It's worth mentioning that in previous studies of the reject inference problem, the lending club data set was the most commonly used data set (Li et al. 2017; Tian et al. 2018; Kim and Cho 2019; Xia et al. 2018; Xia 2019; Anderson 2019; Mancisidor et al. 2020; Liu et al. 2020).

4.2 Performance measures

We use four evaluation measures relevant to credit scoring studies, to assess the performance of our proposed model and benchmarks. These measures are accuracy,

Table 3 Summary of Lending Club numerical data descriptive statistics

	Accepted			Rejected		
	loan_amnt	dti	fico_score	loan_amnt	dti	fico_score
Mean	14601.11	18.12	698.12	15315.66	32.32	675.99
Std	8746.53	9.56	31.66	10786.62	49.51	37.81
Min	500.00	−1.00	627.00	1000.00	−1.00	627.00
25%	8000.0	11.76	672.00	6000.00	11.79	648.00
50%	12075.0	17.52	692.00	12000.00	24.32	668.00
75%	20000.0	23.91	712.00	24000.00	39.44	696.00
Max	40000.00	999.00	847.50	40000.00	998.46	990.00

Table 4 Summary of Lending Club categorical data descriptive statistics

	Accepted			Rejected		
	emp_length	Purpose	addr_state	emp_length	Purpose	addr_state
Unique	11	14	51	11	14	51
Top	10+ years	debt_c	CA	< 1 year	debt_c	CA
Freq	442197	737561	186319	1841718	1067642	266085

Table 5 Binning summary of Lending Club data

name	dtype	n_bins	iv	js	gini	quality_score
loan_amnt	Numerical	9	0.043861	0.005462	0.115813	0.058768
emp_length	Categorical	8	0.001542	0.000192	0.021640	0.000022
Purpose	Categorical	4	0.017069	0.002130	0.061679	0.045264
addr_state	Categorical	10	0.015717	0.001959	0.066421	0.047441
dti	Numerical	15	0.073093	0.009077	0.15316	0.274635
fico_score	Numerical	13	0.128666	0.015747	0.188358	0.500633

precision, recall, and area under the roc curve (AUC). The accuracy evaluates the correctness of label prediction while precision, recall and AUC measure the models discriminative capability. Using the credit scoring model prediction results, summarized in table called confusion matrix (Table 6), the aforementioned measures are computed as follows :

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

. Defined as the proportion of correctly predicted instances to the total number of instances.

$$Precision = \frac{TP}{TP + FP}$$

Table 6 Confusion matrix for the credit scoring domain

	Predicted	
Observed	Good	Bad
Good	(TP) True positive instances	(FN) False negative instances
Bad	(FP) False positive instances	(TN) True negative instances

. Which quantifies the fraction of the predicted positive instances which are true positive.

$$Recall = \frac{TP}{TP + FN}$$

. Which quantifies the number of the predicted positive instances made out of the total number of positive instances.

AUC reflects a classifier's overall behaviour independently of classification threshold values. The model is considered to have a good discriminative capability when its AUC value approaches 1. In contrast, a model is considered to have less efficient discriminative capability when its AUC approaches the value of 0.5. The AUC can be computed as follows :

$$\text{AUC} = \frac{\sum_{i \in \mathcal{M}^+} \sum_{j \in \mathcal{M}^-} \mathbb{1}_{f(\boldsymbol{x}^{(i)}) > f(\boldsymbol{x}^{(j)})}}{N^+ N^-}$$

where $\{f(x) = 0 \mid x \in \mathbb{R}^n\}$ is the separation surface and $\mathbb{1}$ is the indicator function.

4.3 Statistical tests of significance

In the literature, parametric and nonparametric significance tests have been conducted to determine whether one model is significantly better than another. The assumptions of parametric tests, such as normality or homogeneity of variance, are generally broken in practice (Lessmann et al. 2015). Therefore, nonparametric tests are often preferred to parametric tests (Demsar 2006; García et al. 2010). Friedman's test (Friedman 1940) is used in this study to determine if there is a significant difference between models for a certain assessment metric. Although, Friedman aligned rank test and Quade tests are two alternatives to the Friedman test (García et al. 2010), these two tests are favourites over the Friedman test ,if only the compared algorithms are not more than 4 or 5.

The Friedman statistic is computed as follows:

$$\chi_F^2 = \frac{12N}{k(k+1)} \left[\sum_j R_j^2 - \frac{k(k+1)^2}{4} \right], R_j = \frac{1}{N} \sum r_i^j$$

k denotes the number of models, N the number of data samples, R_j the average rank of the j-th model over all the data samples, and r_i^j the j^{th} of k models on the i^{th} of N data samples.

If Friedman's test rejects the null hypothesis of equivalence of ranks for a given evaluation measure, we perform pairwise comparisons using the post-hoc Nemenyi test (Nemenyi 1962) by computing the critical difference (CD):

$$CD = q_\alpha \sqrt{\frac{k(k+1)}{6N}}$$

The crucial values for q_α are based on the studentized range statistic. The results of the Nemenyi post-hoc test are illustrated by a critical distance diagrams, which display the model ranks as well as the critical difference. A horizontal bar connects models that are not significantly different.

Furthermore, for each evaluation measure, we use a Wilcoxon rank-sum test to compare the control approach (the proposed SSHMM model in this research) to a set of benchmark models. This test is more powerful than the post-hoc test, which is used to determine whether a new approach is superior to existing ones (Demsar 2006).

4.4 Experimental design

Our experimental process for evaluating the effectiveness of our proposed framework is described in Fig. 2. We conduct two different sets of experiments. In the first experimental setup Two sets of experiments are performed. In the initial experimental setting, we compare the performance of the SSHMM model with a range of semi-supervised learning techniques for reject inference, including semi-supervised SVM (S3VM), SVM in combination with self-learning, contrastive pessimistic likelihood estimation (CPLE) and label propagation frameworks, also Lightgbm as base classifier with the self-learning and CPLE frameworks. To measure the marginal gain of reject inference, we use a total of six widely used supervised machine learning classifiers in credit scoring (Lessmann et al. 2015): multi-layer perceptron (MLP), support vector machines (SVM), random forest (RF), extreme gradient boosting (XGBoost), light gradient boosting machines (LightGBM), and Categorical Boosting (Catboost). In the second experiment, we change the size of the rejected sample while keeping the size of the accepted sample the same to see how the rejected sample size affects the SSHMM model's predictive ability.

As suggested by Li et al. (2017); Tian et al. (2018); Xia et al. (2018); Xia (2019), the experiment is carried out as follows:

Step 1: Randomly select a sample of accepts and rejects, which sizes are denoted respectively as NA and NR.

Step 2: Randomly divide the accepted samples into a training set and a test set, using the proportion 70%:30%,. Then we choose the number of reject applications to be merged with the training sample, denoted as NR.

Step 3: Respectively build supervised models using the training sample with known labels and semi-supervised models using the training sample (labelled and unlabelled).

Step 4: Predict the likelihood of default and the labels of the test set sample using the classification rules generated in step 3.

Step 5: Compute and compare the model's performance metrics.

Steps 2 through 5 were repeated 25 times, and the evaluation metrics were computed by averaging the results values. Moreover, in the first experiment we set NA to 2000 and we keep the original acceptance ratio 8%. In the second experiment, we set up two alternative scenarios and compared roc curves and AUC scores to see how the rejection rate affects the SSHMM model's performance. We started by setting NA to 2000 and NR to a range of 1000 to 25000. Then we set NA to 1;266;782 and NR to a range of numbers between 1000 and 2;064;314. That's more data to what S3VM can handle due to memory requirements and not feasible for the CPLE procedure due to computing time.

Furthermore, to prevent only considering the accepted cases in the test sample used for models' evaluations, the previous set of experiments were also performed by including the same proportion of rejected cases in the test sample. Thus, the test sample will contain both the accepted and rejected cases (unbiased test sample).

Since the true labels of rejects is unknown, direct estimation of performance is prohibited. Thus, we approximately generate a ground truth for the good/bad label of the rejected cases following the method conducted by Li et al. (2017). It's worth mentioning that just a few research had access to a data set that included a fraction of the rejected applicants data with known outcomes (Kozodoi et al. 2019; Shen et al. 2020). Resulting, e.g. from executing risky strategies as accepting some rejected applicants by the scoring system. As a result, the true repayment status of those applicants who were initially rejected will be known. Unfortunately, the data sets from those studies are private.

4.5 Hyper-parameters settings

Machine learning algorithms have several hyper-parameters that largely influence performance. Thus, we must tune hyper-parameters of these models. We used a grid search with 10-fold cross-validation method to search for the optimal hyper-parameters for SVM, RF, XGBoost, CatBoost, LightGBM, and MLP classifiers. Table 7 shows the summary of the hyper-parameters search space for each of those classifiers. The hyper-parameter optimization in our proposed SSHMM framework is done for the tuning of contamination parameters in the filtering stage, we selected values between 0.01,0.03, 0.05, 0.1 and 0.2. There are various hyper-parameters in machine learning algorithms that have a significant impact on performance. As a result, we must fine-tune these models hyper-parameters. To find the best hyper-parameters for SVM, RF, XGBoost, CatBoost, LightGBM, and MLP classifiers, we performed a grid search with a 10-fold cross-validation approach. For each of the classifiers, Table 7 summarizes the hyper-parameters search space. We select contamination values between 0.01,0.03, 0.05, 0.1, and 0.2 in the filtering stage of suggested SSHMM framework.

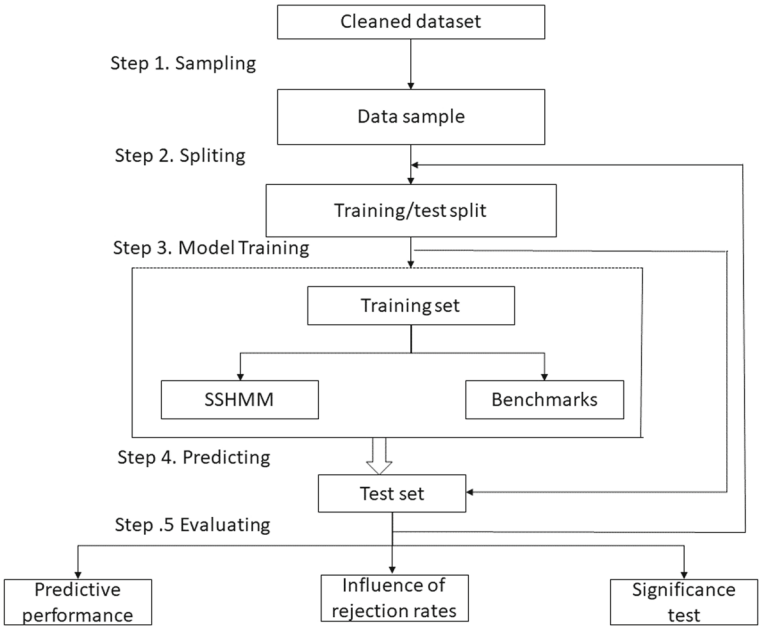


Fig. 2 Flowchart of the experiments set-up

4.6 Results and discussion

Predictive performance analysis

Table 8 shows the numerical experimental results of the proposed SSHMM model and the benchmark models while preserving the original acceptance ratio. The best results for each performance metric, which includ accuracy, precision, recall, and AUC, are highlighted in bold font.

The performance results of the proposed SSHMM model and the benchmark models on the biased test set shows that SSHMM outperform other classifiers over most evaluation measures, namely accuracy, precision and AUC. Particularly, SSHMM improved the classification capability of the base model HMM for the aforementioned evaluation measures. Over all evaluation measures, the S3VM model performed worse than the standard SVM model. and when SVM was combined with self learning, CPLE and label propagation frameworks, the predictive performance deteriorated as well.

The performance results of the proposed SSHMM model and the benchmark models on the unbiased test set shows that MLP model yield the best performance in terms of accuracy, recall and AUC. Lightgbm was the second best model with the same performance as the MLP in terms of recall and AUC. Our proposed SSHMM model was the third best, with the highest precision achieved. Note that, SSHMM improved the classification capability of HMM on all the evaluation measures. We mention that S3VM and self learning frameworks also improved the classification capability of the base model SVM on all the evaluation measures.

Table 7 Grid for hyper-parameters optimization

Method	Hyper-parameters	Search Space
SVM	Gamma	0.0001, 0.001, 0.01, 0.1, 1
	C	0.01, 0.1, 1, 10, 100, 1000
RF	Number estimators	20,50,100,200,300
	Maximum depth	2, 4, 6, 8, 10, 14
	Minimum samples leaf	3, 6, 9, 12, 15, 18, 21
	Minimum samples split	2,5,6,7,9,10
XGBoost	Learning rate	0.006,0.01,0.03,0.1,0.3
	Maximum depth	2, 4, 6, 8, 10, 14
	Number estimators	20,50,100,200,300
	Minimum child weight	1, 2, 3, 4, 5, 8
	Subsample ratio	0.2,0.3,0.5,0.7,0.9
	Colsample by tree ratio	0.5,0.6,0.7,0.8,0.9
LighGBM	Learning rate	0.006,0.01,0.03,0.1,0.3
	Maximum depth	2, 4, 6, 8, 10, 14
	Number estimators	20,50,100,200,300
	Number leaves	30,60,90,100,200
	Bagging fraction	0.3,0.5,0.7,0.8,0.9,1
	Feature fraction	0.3,0.5,0.7,0.8,0.9,1
CatBoost	Learning rate	0.006,0.01,0.03,0.1,0.3
	Maximum depth	2, 4, 6, 8, 10, 14
	Number estimators	20,50,100,200,300
	Random strength	0.2,0.5,0.8
	Bagging temperature	0.03,0.09,0.25,0.75
MLP	Hidden layer sizes	(50,50,50), (50,100,50), (100,),(50,50,100)
	Activation	tanh, relu
	Solver	sgd, adam
	Learning rate	constant, adaptive
	Alpha	0.0001, 0.001, 0.01, 0.15, 0.3

Analysis of signification tests

Friedman test statistics on accuracy, recall, precision, and AUC metrics are presented in Table 8. The Friedman test's null hypothesis is rejected at a 95% level of significance, resulting in significant differences between the different models. We use the Nemenyi post-hoc test to see if there were any significant differences between the models. If the difference in the mean ranks is more than the critical distance, the differences are significant. The results of the post-hoc tests are shown in Figs. 3 and 4. At the 95% level of significance, the models within the bold line are not statistically different.

Furthermore, Table 9 shows the results of significance test on the AUC of the control method SSHMM and the benchmark models using the Wilcoxon Rank-sum

Table 8 Experimental results of models performance comparaison

Model	Evaluation on biased test set				Evaluation on unbiased test set			
	Accuracy	Recall	Precision	AUC	Accuracy	Recall	Precision	AUC
MLP	0.5799	0.6079	0.5701	0.6069	**0.7224**	**0.7989**	**0.7499**	**0.7726**
SVM	**0.5949**	**0.6141**	**0.5874**	**0.6276**	0.5846	0.5945	0.6597	0.6070
Lightgbm	0.5863	0.6117	0.5768	0.6188	**0.7075**	**0.7919**	0.7302	**0.7717**
CatBoost	0.5927	**0.6295**	0.5814	0.6259	0.7003	**0.7927**	0.7232	0.7588
Xgboost	0.5690	**0.6174**	0.5586	0.5984	0.6145	0.7155	0.6567	0.6403
RF	0.5846	0.6033	0.5766	0.6160	0.6878	0.7774	0.7149	0.7463
HMM	**0.5952**	0.5867	**0.5936**	**0.6332**	0.6967	0.7116	**0.7574**	0.7554
SSHMM	**0.6140**	0.5810	**0.6186**	**0.6499**	**0.7096**	0.7293	**0.7646**	**0.7624**
S3VM	0.5650	0.5666	0.5648	0.5647	0.6814	0.6910	0.7437	0.6815
SelfLearning SVM	0.5639	0.5665	0.5580	0.5850	0.5920	0.6005	0.6669	0.6207
CPLE SVM	0.5480	0.4328	0.5534	0.5570	0.4433	0.2593	0.5485	0.4296
Label Propagation SVM	0.5445	0.5260	0.5400	0.5553	0.5580	0.5598	0.6380	0.5724
SelfLearning Lightgbm	0.5825	0.6089	0.5736	0.6078	0.6106	0.6686	0.6654	0.6376
CPLE Lightgbm	0.5628	0.5788	0.5547	0.5798	0.4294	0.3451	0.5164	0.4016
Friedman test statistic	6.9560	3.1956	5.7781	8.8282	3.0116	5.3760	13.841	3.6802

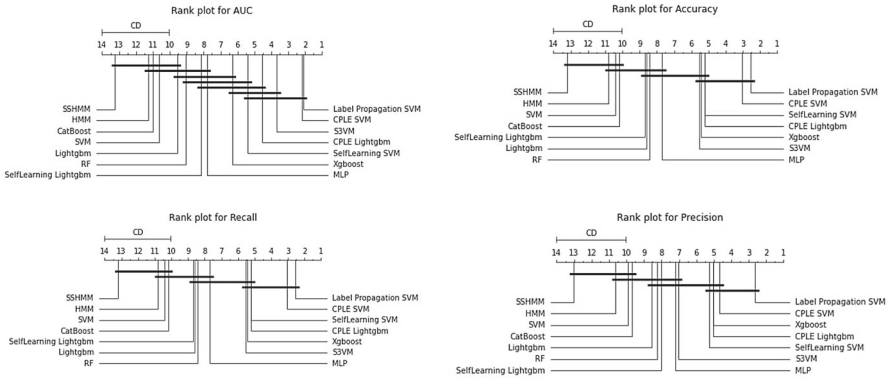


Fig. 3 CD diagrams of Nemenyi post-hoc tests on the biased test set

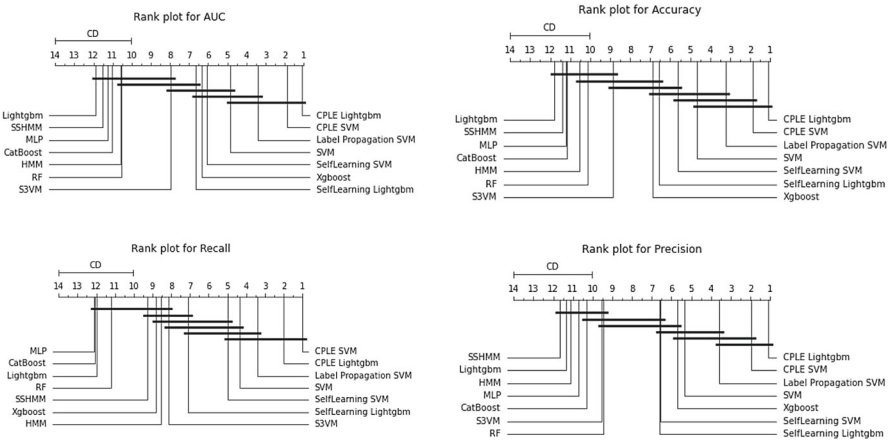


Fig. 4 CD diagrams of Nemenyi post-hoc tests on the unbiased test set

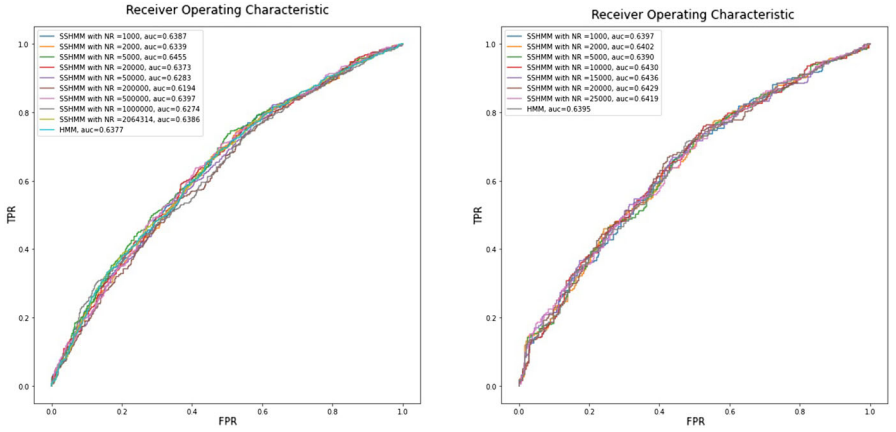
test. The significance level of the tests is alpha=0.05. The null hypothesis of the tests is "There is no significant difference between AUC performance of the control model SSHMM and AUC of the model used as comparison". Subsequently, SSHMM is significantly better than benchmark models on AUC performance over the biased test set (p-value < 0.05). However, the p-value calculated between SSHMM and MLP, Lightgbm, Catboost, RF and HMM models were greater than 0.005 which indicates a statistically insignificant differences over the unbiased test set. Consequently, The results highlight the efficiency of the proposed model.

Analysis of rejection rate influence

To investigate the impact of rejection rates on AUC performance and to identify the optimal rejection rate for the SSHMM, we randomly sampled rejected data set with different rejection rates. The ROC curves in Fig. 5 lead to the following conclusions. First, the results show that the proposed SSHMM model can reach optimal performance

Table 9 Wilcoxon Rank-sum test on AUC mesure

Control Model	Benchmark Models	P value biased test set	P value unbiased test set
SSHMM	HMM	0.0065	0.5343
	RF	0.001	0.195
	CatBoost	0.001	0.7434
	MLP	0.0001	0.5408
	Xgboost	0.001	0.001
	CPLE SVM	0.001	0.001
	SVM	0.0005	0.001
	S3VM	0.001	0.001
	SelfLearning SVM	0.001	0.001
	Label Propagation SVM	0.0001	0.0001
	Lightgbm	0.001	0.2948
	SelfLearning Lightgbm	0.001	0.001
	CPLE Lightgbm	0.001	0.001



(a) NA=1,266,782 and NR= 1000 to 2,064,314

(b) NA=2000 and NR= 1000 to 25000

Fig. 5 ROC curves of SHHMM model under different rejection rates

without requiring a large number of rejected samples. It also shows that using samples with a low rejection rate improves predictive accuracy than those using samples with a higher rejection rate. Second, when the rejection rate increased, the SSHMM's predictive performance changed, but in most circumstances, the SSHMM's ROC curves outperformed the supervised HMMs.

Conclusion

In terms of semi-supervised learning, there have been few successful methodologies to solve the problem of reject inference in the credit scoring domain. Using a semi-supervised modified HMM model, this study offers a novel approach to the problem. The SSHMM model outperforms other models in terms of applicability, stability, and performance when tested on real P2P lending data. More crucially, by using the prospective information of rejected candidates, the prediction performance of the underlying HMM classifier has improved using the suggested framework. We can look into the following directions for future research. First, because the Baum–Welch algorithm is recognized to convergence towards local optimums, we can use different algorithms to estimate HMM parameters (El annas et al. 2022). Second, we can consider building an ensemble method incorporating the existing machine learning methods together with SSHMM to do the reject inference.

References

Anderson R (2007) The credit scoring toolkit: theory and practice for retail credit risk management and decision automation. Oxford University Press, Oxford

Anderson B (2019) Using Bayesian networks to perform reject inference. Expert Syst Appl 137:349–356

Banasik J, Crook J (2007) Reject inference, augmentation, and sample selection. Eur J Oper Res 183(3):1582–1594

Banasik J, Crook J (2010) Reject inference in survival analysis by augmentation. J Oper Res Soc 61(3):473–485

Banasik J, Crook J, Thomas LC (2003) Sample selection bias in credit scoring models. JORS 54(8):822–832

Baum LE, Petrie T, Soules G, Weiss N (1970) A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann Math Stat 41:164–71

Bücker M, van Kampen M, Krämer W (2013) Reject inference in consumer credit scoring with nonignorable missing data. J Bank Finance 37(3):1040–1045

Chen GG, Astebro T (2001) The economic value of reject inference in credit scoring. Department of Management Science, University of Waterloo, Waterloo

Crook J, Banasik J (2004) Does reject inference really improve the performance of application scoring models? J. Bank Finance 28(4):857–874

Demsar J (2006) Statistical comparisons of classifiers over multiple datasets. J Mach Learn Res 7:1–30

El annas M, Ouzineb M, Benyacoub B (2022) Hidden Markov models training using hybrid Baum Welch: variable neighborhood search algorithm. Stat Optim Inf Comput 10(1):160–170

Feelders AJ (1999) Credit scoring and reject inference with mixture models. Intell Syst AccountFinance Manag 8:271–279

Friedman M (1940) A comparison of alternative tests of significance for the problem of *m* rankings. Ann Math Stat 11(1):86–92

García S, Fernández A, Luengo J, Herrera F (2010) Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power. Inf Sci 180:2044–2064

https://home.kpmg/xx/en/home/insights/2020/02/pulse-of-fintech-archive.html

https://www.lendingclub.com/info/download-data.action

Kang Y, Jia N, Cui R, Deng J (2021) A graph-based semi-supervised reject inference framework considering imbalanced data distribution for consumer credit scoring. Appl Soft Comput 105:107259

Kim A, Cho S-B (2019) An ensemble semi-supervised learning method for predicting defaults in social lending. Eng Appl Artif Intell 81:193–199

Kozodoi N, Katsas P, Lessmann S, Moreira-Matias L, Papakonstantinou K (2019). Shallow self-learning for reject inference in credit scoring. In: Joint European conference on machine learning and knowledge discovery in databases, pp 516–532. Springer

Lessmann S, Baesens B, Seow HV, Thomas LC (2015) Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research. Eur J Oper Res 247:124–136
Levinson SE, Rabiner LR, Sondhi MM (1983) An introduction to the application of the theory of probabilistic functions of Markov process to automatic speech recognition. The Bell Syst Tech J 62:1035–74
Li X, Parizeau M, Plamondon R (2000) Training hidden Markov models with multiple observations-a combinatorial method. IEEE Trans Pattern Anal Mach Intell 22:371–77
Li Z, Tian Y, Li K, Zhou F, Yang W (2017) Reject inference in credit scoring using Semi-supervised support vector machines. Expert Syst Appl 74:105–114
Liu FT, Ting KM, Zhou ZH (2008) Isolation forest. In: 2008 eighth IEEE international conference on data mining. pp 413–422. IEEE
Liu Y, Li X, Zhang Z (2020) A new approach in reject inference of using ensemble learning based on global semi-supervised framework. Futur Gener Comput Syst 109:382–391
Maldonado S, Paredes G (2010) A semi-supervised approach for reject inference in credit scoring using svms. In: Industrial conference on data mining. pp 558–571. Springer
Mancisidor RA, Kampffmeyer M, Aas K, Jenssen R (2020). Deep generative models for reject inference in credit scoring. Knowl-Based Syst, 105758
Marshall A, Tang L, Milne A (2010) Variable reduction, sample selection bias and bank retail credit scoring. J Empir Financ 17(3):501–512
Navas-Palencia G (2020) Optimal binning: mathematical programming formulation. http://arxiv.org/abs/2001.08025
Nemenyi P (1962) Distribution-free multiple comparisons. In: Biometrics, Vol. 18, international biometric Soc 1441 I ST, NW, SUITE 700, Washington, DC 20005-2210, p 263
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. Decis Supp Syst 137:113366
Siddiqi N (2017) Intelligent credit scoring: building and implementing better credit risk scorecards, 2nd edn. Wiley, Hoboken, NJ
Sohn S, Shin S (2006) Reject inference in credit operations based on survival analysis. Expert Syst Appl 31(1):26–29
Tian Y, Yong Z, Luo J (2018) A new approach for reject inference incredit scoring using kernel-free fuzzy quadratic surface support vector machines. Appl Soft Comput 73:96–105
Xia Y (2019) A novel reject inference model using outlier detection and gradient boosting technique in peer-to-peer lending. IEEE Access 7:92893–92907
Xia Y, Yang X, Zhang Y (2018) A rejection inference technique based on contrastive pessimistic likelihood estimation for P2P lending. Electron. Commerce Res. Appl. 30:111–124