Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Polynomial Classification EM for Reject Inference in P2P Credit Scoring: Semi-Supervised Learning Under Non-Random Selection

Abderrahim EL AMRANI
Institut National de Statistique et d'Economie Appliquée
Badreddine BENYACOUB
Institut National de Statistique et d'Economie Appliquée
Mohammed EL HAJ TIRARI
Institut National de Statistique et d'Economie Appliquée
Ismail El Moudden
elmoudi@odu.edu
Old Dominion University

Research Article

Keywords: Credit scoring, Reject inference, Semi-supervised learning, P2P lending, Classification EM, Polynomial features

Posted Date: April 24th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9098066/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Polynomial Classification EM for Reject Inference in P2P Credit Scoring: Semi-Supervised Learning Under Non-Random Selection

Abderrahim EL AMRANI[1], Badreddine BENYACOUB[2], Mohammed EL HAJ TIRARI[3], Ismail EL MOUDDEN[4*]

[1,2,3]Research Laboratory in Applied Methods in Statistics, Actuarial Science, Finance and Quantitative Economics, INSEA, Avenue Allal El Fassi, P.O. Box 6217, Rabat, Morocco.
[4]Old Dominion University, Norfolk, VA, USA.

*Corresponding author(s). E-mail(s): elmoudi@odu.edu;
Contributing authors: aelamrani@insea.ac.ma;
bbenyacoub@insea.ac.ma; tirari@insea.ac.ma;

Abstract

Credit scoring models trained exclusively on accepted applicants suffer from sample selection bias, a problem amplified in peer-to-peer (P2P) lending where rejection rates exceed 90%. Reject inference methods attempt to correct this bias by incorporating information from rejected applicants, yet most approaches assume linear decision boundaries and rarely evaluate whether the underlying missing-data assumptions hold empirically.

We propose LogisticCEMD2, a semi-supervised classification algorithm that integrates degree-2 polynomial feature expansion with $\boldsymbol{\ell_1}$ regularization within the Classification Expectation-Maximization (CEM) framework. The polynomial extension captures pairwise feature interactions and quadratic effects while $\boldsymbol{\ell_1}$ penalization performs automatic feature selection among the expanded feature set. We develop a mini-batch variant with proven convergence guarantees and $\boldsymbol{\mathcal{O}(Jd^2(B + In_l))}$ computational complexity.

We evaluate the method on LendingClub data comprising 1,266,782 accepted and 2,401,300 rejected loan applications (2007–2018). Empirical validation reveals that the selection mechanism is strongly non-random (classification AUC = 0.943), establishing that the Missing at Random assumption underlying CEM is substantially violated. All subsequent results therefore characterize CEM behavior under documented Missing Not at Random conditions. Across 25 independent

runs comparing 12 classifiers, Friedman tests confirm significant performance differences ($\chi^2 = 193.4$, $p < 0.001$). With the standard hard classification step, LogisticCEMD2 achieves AUC 0.602 with particular strength in recall (0.656), but exhibits poor calibration (Brier = 0.304). An ablation study with Bonferroni-corrected Wilcoxon signed-rank tests reveals that replacing the hard classification step with soft posterior weights (SoftEM) recovers both discrimination (AUC = 0.633, matching the supervised baseline) and calibration (Brier = 0.236; Hosmer-Lemeshow $p = 0.083$), demonstrating that the discrimination loss is caused by the binarization of posteriors rather than the CEM framework itself. Sensitivity analysis reveals performance degradation when unlabeled samples substantially exceed labeled samples.
These results establish that polynomial CEM with soft assignment preserves both discrimination and calibration under MAR violation, while the hard C-step variant trades discrimination for sensitivity. Practical value depends on the label assignment strategy and the asymmetric cost structure typical of credit scoring.

Keywords: Credit scoring, Reject inference, Semi-supervised learning, P2P lending, Classification EM, Polynomial features

1 Introduction

Peer-to-peer (P2P) lending platforms have expanded access to consumer credit by connecting borrowers directly with investors, bypassing traditional banking intermediaries. LendingClub, the largest U.S. platform, facilitated $1.85 billion in loan originations in 2024, a 13% increase over the prior year (LendingClub Corporation 2024). This growth creates a corresponding need for accurate credit risk assessment: even marginal improvements in default prediction translate to measurable reductions in portfolio loss for both platforms and investors (Hand and Henley 1997).

A persistent challenge in credit scoring is sample selection bias. Scoring models are typically trained on accepted applicants only, because default outcomes are observed exclusively for borrowers who received loans (Thomas 2000; Baesens et al. 2003). In the LendingClub dataset spanning 2007–2018, only 8% of applicants were accepted, meaning the training sample excludes 92% of the applicant population. When the acceptance decision depends on the same features used for scoring, the resulting model is biased toward the characteristics of accepted borrowers and may perform poorly on the full applicant distribution (Heckman 1979; Feelders 2000). This problem is formally known as reject inference: estimating default risk for applicants whose outcomes are unobserved because their applications were denied. Missing data mechanisms (Heitjan and Basu 1996) determine the theoretical properties of reject inference approaches: if selection is Missing at Random (MAR), the labeled sample is informative about the full population conditional on observed features, whereas Missing Not at Random (MNAR) invalidates most standard correction methods.

Several approaches to reject inference have been proposed. Reweighting adjusts the contribution of each accepted applicant based on its probability of selection (Bücker et al. 2013). Augmentation assigns pseudo-labels to rejected applicants and retrains

the model (Banasik et al. 2003). Semi-supervised methods treat rejected applicants as unlabeled data and estimate their default probabilities jointly with model parameters (Li et al. 2017; Chapelle et al. 2006). Among these, the Classification Expectation-Maximization (CEM) framework combines a discriminative classifier with iterative label assignment and parameter re-estimation (Celeux and Govaert 1992; Vittaut et al. 2002; Amini and Gallinari 2002). Anderson and Hardin (2013) demonstrated that logistic CEM improves credit scoring when applied to reject inference. El Annas et al. (2023) proposed an alternative semi-supervised framework using Hidden Markov Models on LendingClub data, while Shen et al. (2021) developed deep learning approaches for credit scoring with oversampling. Ensemble methods (Wang et al. 2011; West et al. 2005; Wang et al. 2012; Lessmann et al. 2015) and SVM-based approaches (Bellotti and Crook 2009) have shown strong performance in credit scoring benchmarks. However, existing CEM-based approaches rely on linear logistic regression, limiting the decision boundary to a hyperplane and ignoring feature interactions (Murty and Raghava 2016).

In this paper, we extend the semi-supervised logistic CEM framework in three ways. First, we introduce a degree-D polynomial feature expansion that enables non-linear decision boundaries through quadratic terms and pairwise interactions, specializing to $D = 2$ (LogisticCEMD2) to balance expressiveness against the curse of dimensionality. Second, we incorporate ℓ_1 regularization to perform automatic feature selection among the expanded polynomial feature set, reducing the effective dimensionality from $\mathcal{O}(d^2)$ to a sparse subset. Third, we develop a mini-batch variant that processes random subsets of the unlabeled data in each EM iteration, with a scaling factor that yields an unbiased estimator of the full-batch objective and convergence guarantees as the batch size approaches the full unlabeled sample. Our specific aims are: (1) to quantify the degree of Missing at Random (MAR) violation in LendingClub data and characterize its impact on CEM-based reject inference; (2) to compare LogisticCEMD2 against 12 supervised and semi-supervised baselines across 25 independent runs with formal statistical tests; (3) to quantify the individual and joint contributions of each algorithmic component via ablation; and (4) to characterize model sensitivity to the volume of unlabeled data.

2 Methods

Reject inference maps directly onto the semi-supervised learning (SSL) framework: accepted applicants with observed outcomes constitute the *labeled* dataset, rejected applicants whose default status is unobserved form the *unlabeled* dataset, the platform's acceptance decision functions as the *selection mechanism*, and the reject inference objective (estimating default risk for the full applicant population) is equivalent to the SSL classification objective of learning from both labeled and unlabeled data (Chapelle et al. 2006). The key question is whether the selection mechanism satisfies the Missing at Random (MAR) condition (Little and Rubin 2019), i.e., whether acceptance depends only on observed features and not on the unobserved outcome. When MAR is violated, the labeled sample is not representative of the full population

conditional on features alone, and semi-supervised methods that implicitly assume MAR may produce biased estimates.

2.1 Problem Formulation

Let $D_l = \{(\mathbf{x}_i, z_i)\}_{i=1}^{n_l}$ denote the labeled dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector and $z_i \in \{0, 1\}$ the binary class label (0 = non-default, 1 = default). Let $D_u = \{\mathbf{x}_i\}_{i=n_l+1}^{n_l+m}$ denote the unlabeled dataset of m rejected applicants. Under the cluster assumption in semi-supervised learning (Zhu and Goldberg 2009), points within the same cluster are likely to share the same class label.

The complete-data log-likelihood for the combined dataset is:

$$LL_c = \sum_{i=1}^{n_l} \sum_{k=1}^{2} z_{ik} \log P(P_k \mid \mathbf{x}_i, \boldsymbol{\theta}) + \sum_{i=n_l+1}^{n_l+m} \sum_{k=1}^{2} z_{ik} \log P(P_k \mid \mathbf{x}_i, \boldsymbol{\theta}), \tag{1}$$

where $z_{ik} \in \{0, 1\}$ is the indicator that observation i belongs to class k, P_k denotes class k, and $\boldsymbol{\theta} = \boldsymbol{\beta}$ collects the model parameters. For labeled data, z_{ik} is observed directly from the outcome; for unlabeled data, it is estimated during the E-step. For binary classification ($K = 2$), the class indicators satisfy $z_{i1} + z_{i2} = 1$, and we define the scalar shorthand $z_i \equiv z_{i1} \in \{0, 1\}$ (with $z_{i2} = 1 - z_i$). This binary shorthand z_i is used in all subsequent notation, including Algorithm 1.

Using a discriminative formulation (Vittaut et al. 2002), the posterior probability is modeled as:

$$P(P_1 \mid \mathbf{x}_i, \boldsymbol{\theta}) = \sigma(\boldsymbol{\beta}^\top \mathbf{x}_i) = \frac{1}{1 + \exp(-\boldsymbol{\beta}^\top \mathbf{x}_i)}, \tag{2}$$

where $\sigma(\cdot)$ is the logistic function and $\boldsymbol{\beta}$ includes an intercept term.

2.2 Polynomial Extension

Standard logistic CEM restricts the decision boundary to a hyperplane. To capture non-linear relationships, we replace $\mathbf{x}_i$ with a polynomial feature map $\boldsymbol{\phi}_D(\mathbf{x}_i)$ of degree D:

$$\log\left(\frac{\pi_i}{1 - \pi_i}\right) = \beta_0 + \sum_{\substack{\alpha \in \mathbb{N}^d \\ 1 \le |\alpha| \le D}} \beta_\alpha \mathbf{x}^\alpha, \tag{3}$$

where $\alpha = (\alpha_1, \ldots, \alpha_d)$ is a multi-index with $|\alpha| = \sum_j \alpha_j$, and $\mathbf{x}^\alpha = \prod_j x_j^{\alpha_j}$.

For $D = 2$ with d input features, the expanded feature vector $\boldsymbol{\phi}_2(\mathbf{x}_i) \in \mathbb{R}^p$ has dimension:

$$p = d + d + \frac{d(d-1)}{2} = 2d + \frac{d(d-1)}{2}, \tag{4}$$

comprising d linear terms, d squared terms, and $\binom{d}{2}$ pairwise interaction terms. For $d = 8$ features, this yields $p = 44$ polynomial features.

Algorithm 1 LogisticCEMD2

Require: Labeled data D_l, unlabeled data D_u, degree $D = 2$, regularization λ, tolerance ϵ, max iterations $J_{\max}$, patience P

Compute $\mathbf{\Phi}_l = \boldsymbol{\phi}_2(D_l)$, $\mathbf{\Phi}_u = \boldsymbol{\phi}_2(D_u)$
Train initial model: $G^{(0)} \leftarrow \text{LogisticRegression}(\ell_1, C = 1/\lambda).\text{fit}(\mathbf{\Phi}_l, \mathbf{y}_l)$
$j \leftarrow 0$, best_ll $\leftarrow -\infty$, no_improve $\leftarrow 0$
while $j < J_{\max}$ **do**
    E-step: $p_i^{(j)} \leftarrow G^{(j)}.\text{predict_proba}(\mathbf{\Phi}_u)$ ▷ Posterior probabilities
    C-step: $z_i^{(j+1)} \leftarrow \mathbb{1}[p_i^{(j)} > 0.5]$ for all $\mathbf{x}_i \in D_u$ ▷ Hard label assignment
    M-step: $G^{(j+1)} \leftarrow \text{LogisticRegression}(\ell_1, C = 1/\lambda).\text{fit}(\mathbf{\Phi}_l \cup \mathbf{\Phi}_u, \mathbf{y}_l \cup \mathbf{z}^{(j+1)})$
    Compute $\widehat{LL}_c^{(j+1)}$ via Eq. (5)
    Early stopping: if $|\widehat{LL}_c^{(j+1)} - \widehat{LL}_c^{(j)}| < \epsilon$ for P consecutive iterations, **break**
    $j \leftarrow j + 1$
end while
return $G^{(\text{best})}$ (model with highest $\widehat{LL}_c$)

2.3 LogisticCEMD2 Algorithm

Adding ℓ_1 regularization to the complete-data log-likelihood for joint feature selection and parameter estimation:

$$\widehat{LL}_c = \sum_{k=1}^{2} \sum_{\mathbf{x}_i \in P_k} \log P(P_k \mid \boldsymbol{\phi}_2(\mathbf{x}_i), \boldsymbol{\theta}) + \sum_{k=1}^{2} \sum_{i=n_l+1}^{n_l+m} z_{ik} \log P(P_k \mid \boldsymbol{\phi}_2(\mathbf{x}_i), \boldsymbol{\theta}) - \lambda \|\boldsymbol{\beta}\|_1, \tag{5}$$

where $\lambda > 0$ controls regularization strength.

Algorithm 1 presents the full LogisticCEMD2 procedure, and Algorithm 2 presents its mini-batch extension.

2.4 Convergence of the Mini-batch Estimator

Theorem 1 *Let $\boldsymbol{\beta}_{FB}^{(j)}$ and $\boldsymbol{\beta}_{MB}^{(j)}$ denote the parameter vectors after j iterations of the full-batch and mini-batch algorithms, respectively. The scaled mini-batch objective satisfies:*

1. ***Unbiasedness:*** $\mathbb{E}_{S^{(j)}}[\widetilde{LL}_c^{MB}(\boldsymbol{\beta}^{(j)})] = \widehat{LL}_c^{FB}(\boldsymbol{\beta}^{(j)})$
2. ***Variance bound:*** $\text{Var}(\widetilde{LL}_c^{MB}(\boldsymbol{\beta}^{(j)})) = \frac{m^2\sigma^2(m-B)}{B(m-1)} \leq \frac{m^2\sigma^2}{B}$, *where σ^2 is the finite-population variance of per-sample log-likelihood contributions*
3. ***Convergence:*** $\widetilde{LL}_c^{MB}(\boldsymbol{\beta}^{(j)}) \xrightarrow{p} \widehat{LL}_c^{FB}(\boldsymbol{\beta}^{(j)})$ *as* $B \to m$

When $B = m$, the two algorithms coincide exactly.

Algorithm 2 Mini-batch LogisticCEMD2

Require: As Algorithm 1, plus batch size B

Compute $\mathbf{\Phi}_l = \boldsymbol{\phi}_2(D_l)$, $\mathbf{\Phi}_u = \boldsymbol{\phi}_2(D_u)$
Scale factor: $s = m/B$
Train initial model: $G^{(0)} \leftarrow \text{LogisticRegression}(\ell_1, C = 1/\lambda).\text{fit}(\mathbf{\Phi}_l, \mathbf{y}_l)$
for $j = 0, 1, \ldots, J_{\max} - 1$ **do**
    Sample $S^{(j)} \subset D_u$ uniformly without replacement, $|S^{(j)}| = \min(B, m)$
    E-step: $p_i^{(j)} \leftarrow G^{(j)}.\text{predict_proba}(\boldsymbol{\phi}_2(S^{(j)}))$
    C-step: $z_i^{(j+1)} \leftarrow \mathbb{1}[p_i^{(j)} > 0.5]$ for all $\mathbf{x}_i \in S^{(j)}$
    M-step: Fit $G^{(j+1)}$ on $\mathbf{\Phi}_l \cup \boldsymbol{\phi}_2(S^{(j)})$ with sample weights $w_i = 1$ (labeled), $w_i = s$ (unlabeled)
    Compute $\widetilde{LL}_c^{(j+1)} = LL_l + s \cdot LL_u^{S^{(j)}} - \lambda\|\boldsymbol{\beta}\|_1$
    Early stopping check as in Algorithm 1
end for
return $G^{(\text{best})}$

Proof Under uniform sampling without replacement of B items from m, each $\mathbf{x}_i \in D_u$ satisfies $\mathbb{P}(\mathbf{x}_i \in S^{(j)}) = B/m$. The expected mini-batch log-likelihood contribution is:

$$\mathbb{E}[L_u^{MB}(\boldsymbol{\beta}^{(j)})] = \sum_{\mathbf{x}_i \in D_u} \frac{B}{m} \ell(\mathbf{x}_i, z_i^{(j+1)}, \boldsymbol{\beta}^{(j)}) = \frac{B}{m} L_u^{FB}(\boldsymbol{\beta}^{(j)}).$$

The scaling factor m/B cancels, yielding $\mathbb{E}[\widetilde{LL}_c^{MB}] = \widehat{LL}_c^{FB}$. For the variance, the finite-population correction for sampling without replacement gives:

$$\text{Var}\left(\frac{m}{B} \sum_{\mathbf{x}_i \in S^{(j)}} \ell(\mathbf{x}_i, \cdot)\right) = \frac{m^2\sigma^2(m-B)}{B(m-1)} \leq \frac{m^2\sigma^2}{B},$$

which vanishes as $B \to m$. Convergence in probability follows from Chebyshev's inequality. □

2.5 Computational Complexity

The polynomial expansion produces $p = \mathcal{O}(d^2)$ features. Each EM iteration requires: the E-step at $\mathcal{O}(Bd^2)$ for computing posteriors on the mini-batch, the C-step at $\mathcal{O}(B)$ for thresholding, and the M-step at $\mathcal{O}(I(n_l + B)d^2)$ for fitting the penalized logistic regression with I internal solver iterations. For J EM iterations, the total complexity is $\mathcal{O}(J \cdot (Bd^2 + I(n_l + B)d^2))$, which reduces to $\mathcal{O}(JIn_l d^2)$ when $B \ll n_l$ and is significantly more efficient than the full-batch cost of $\mathcal{O}(JI(n_l + m)d^2)$.

2.6 Data and Preprocessing

We use the LendingClub dataset from Kaggle (Verthez 2017), comprising loan applications from 2007 to 2018. The accepted dataset contains 2,260,701 loans with 150 features; the rejected dataset contains 27,648,741 applications with 6 features. After filtering to terminal loan statuses (Fully Paid = class 0, Charged Off = class 1) and removing records with missing values ($< 3\%$), the accepted dataset contains 1,266,782

Table 1 Descriptive statistics of accepted and rejected loan datasets after preprocessing.

	Accepted (n = 1,266,782)				**Rejected** (n = 2,401,300)			
	Mean	Std	Min	Max	Mean	Std	Min	Max
Loan amount ($)	14,601	8,747	500	40,000	14,343	10,868	1,000	40,000
DTI (%)	18.1	9.6	−1.0	100.0	25.6	19.5	−1.0	100.0
FICO score	698.1	31.7	627.0	847.5	676.2	39.2	627.0	990.0

loans (1,019,356 non-default; 247,426 default). Identical outlier filters are applied symmetrically to both accepted and rejected datasets to ensure consistent feature distributions: DTI $\leq$ 100%, loan amount $\leq$ \$40,000, and FICO $\geq$ 627. We cap DTI at 100% following industry convention; 283,054 rejected observations (3.15%) with DTI $>$ 100% were excluded as likely data entry errors or system placeholders, along with a negligible number of accepted observations exceeding the same thresholds. The final rejected dataset contains 2,401,300 applications.

Since reject inference requires features common to both accepted and rejected datasets (the only features for which both populations are observed), we retain eight features: loan amount, debt-to-income ratio (DTI), FICO score (mean of the range endpoints), employment length (ordinally encoded), loan purpose (label encoded), and three binary indicators for state-level default risk (d1: high, d2: medium, d3: low). The state risk groups are defined once from the full historical accepted dataset and hardcoded in the experimental configuration, representing prior domain knowledge about geographic risk variation rather than a data-driven grouping that would need to respect train/test boundaries. The 150 features available only for accepted applicants are discarded because they cannot be observed for rejected applicants, a constraint inherent to all reject inference approaches. The dataset spans 2007–2018, encompassing the financial crisis, Dodd-Frank Act, and CFPB establishment; we sample uniformly across years without temporal stratification because the year variable is unavailable for rejected applicants.

Continuous features (loan amount, DTI, FICO score) were standardized using a `StandardScaler` fitted on the labeled (accepted) data only, to prevent information leakage from the unlabeled (rejected) distribution into the scaling parameters.

Table 1 presents descriptive statistics for both datasets.

2.7 Experimental Design

All experiments use the implementation described in Section 2.3, with the LIBLINEAR solver (Fan et al. 2008) for ℓ_1-penalized logistic regression. The complete reproducibility package, including the full experimental pipeline, preprocessing scripts, hyperparameter configurations, and all random seeds, is available at https://github.com/isamil/polynomial-cem-credit-scoring.

2.7.1 MAR Validation

We test whether the selection mechanism satisfies the Missing at Random (MAR) assumption by training a logistic regression to classify accepted versus rejected applications using the eight common features. A 500-iteration bootstrap procedure estimates the AUC distribution. The Hosmer-Lemeshow goodness-of-fit test is applied.

2.7.2 Regularization Tuning

Five-fold cross-validation on the labeled data evaluates $\lambda \in \{0.001, 0.01, 0.1, 0.5, 1.0, 2.0, 3.0, 4.0, 5.0, 7.0, 10.0\}$. For each fold, $X_{\text{unlabeled}}$ is passed to the `fit` method. We track training AUC, validation AUC, number of selected features, the overfit metric (train AUC − validation AUC), and the coefficient vectors to identify which features are zeroed out at each regularization level.

2.7.3 Initialization Strategies

Five initialization strategies are compared: Baseline (supervised LR on labeled data), Random (random coefficient initialization), K-means (cluster centroids for initial label assignment), Semi-random (supervised LR with random perturbation), and Conservative (all unlabeled initially assigned to the majority class). Each strategy is run 15 times with different seeds to provide reliable mean and standard deviation estimates.

2.7.4 Model Comparison

Following Anderson and Hardin (2013), who established $N_A = 2{,}000$ as the standard labeled sample size for CEM evaluation, we sample $N_A = 2{,}000$ labeled observations (balanced classes) and $N_R = 25{,}000$ unlabeled rejected applications per run, with a 70/30 train-test split on the labeled data. This sample size is deliberately conservative to test whether CEM can extract value from unlabeled data when labeled data is scarce, the scenario where semi-supervised methods offer the greatest theoretical advantage. Section 2.7.5 tests generalizability at $N_A = 50{,}000$. We compare 12 classifiers: Logistic Regression, SVM, Random Forest, XGBoost, LightGBM, MLP, SuperLearner (stacking ensemble), LogisticCEM (linear, $D = 1$), LogisticCEMD2 ($D = 2$, $\lambda = 5.0$), CEM-XGBoost, PseudoLabel-XGB, and LabelPropagation. All models undergo hyperparameter tuning via grid search (Table 2). We conduct 25 independent runs with distinct random seeds. Performance is evaluated using accuracy, precision, recall, F1 score, and AUC. Statistical significance is assessed using the Friedman test (Demšar 2006) with Nemenyi post-hoc analysis at $\alpha = 0.05$.

2.7.5 Large-Sample Model Comparison

To test whether the classifier ranking generalizes beyond the small-sample setting, we repeat the model comparison at $N_A = 50{,}000$ labeled observations (balanced classes) with $N_R = 25{,}000$ across 10 independent runs. The 25-fold increase in labeled data allows ensemble methods to exploit their capacity advantage. Kernel SVM is replaced with LinearSVC (wrapped in `CalibratedClassifierCV` with sigmoid calibration)

due to the $\mathcal{O}(n^2)$ complexity of kernel methods at this scale; all other models and hyperparameters remain unchanged.

2.7.6 Ablation Study

Nine configurations isolate the contribution of each algorithmic component: (1) Base = Logistic Regression, (2) Base+SSL = Logistic CEM ($D = 1$, no ℓ_1), (3) Base+Poly = Polynomial LR ($D = 2$, no SSL), (4) Base+Poly+ℓ_1 = Polynomial LR with ℓ_1, (5) Base+SSL+Poly = CEM with polynomial features (no ℓ_1), (6) FullModel = LogisticCEMD2 (hard C-step), (7) SelfTraining = sklearn SelfTrainingClassifier with LR base, (8) SoftEM = LogisticCEMD2 with soft posterior weights retaining fractional class responsibilities instead of binarizing at 0.5, and (9) SoftEM+Linear = LogisticCEM ($D = 1$) with soft assignment. Configurations 8–9 test whether the hard C-step (which binarizes posteriors into deterministic pseudo-labels) discards discriminatory information relative to the standard EM approach of retaining probabilistic weights. We conduct 25 independent runs and apply Wilcoxon signed-rank tests for six pre-specified pairwise comparisons with Bonferroni correction ($\alpha_{\text{adj}} = 0.05/6 = 0.0083$): (i) Base vs. Base+SSL, (ii) Base+SSL vs. FullModel, (iii) Base+SSL+Poly vs. FullModel, (iv) SelfTraining vs. FullModel, (v) FullModel vs. SoftEM, and (vi) Base+SSL vs. SoftEM+Linear.

2.7.7 Reject Size Sensitivity

We assess the effect of unlabeled sample size on LogisticCEMD2 under two scenarios, each with 5 independent repeats per N_R value to quantify uncertainty. Scenario 1 fixes $N_A = 2{,}000$ and varies $N_R \in \{1{,}000; 2{,}000; 5{,}000; 10{,}000; 15{,}000; 20{,}000; 25{,}000\}$. Scenario 2 uses the full balanced labeled dataset ($N_A \approx 494{,}852$ after random undersampling of the majority class to match the minority class count; no synthetic oversampling is applied) and varies $N_R \in \{1{,}000; 10{,}000; 100{,}000; 1{,}000{,}000; 2{,}064{,}314\}$.

3 Results

3.1 MAR Validation

The logistic regression classifier distinguishing accepted from rejected applications achieved a bootstrap AUC of 0.943 (95% CI: 0.940–0.945) across 500 iterations (Table 3). The Hosmer-Lemeshow test rejected model calibration ($\chi^2 = 647.6$, $p < 0.001$), indicating poor fit in the tails despite strong discrimination. All eight features contributed to the classification, with state risk indicators (coefficients: -1.29 to -1.24), employment length (0.57), and FICO score (-0.48) showing the largest absolute effects.

An AUC of 0.943 confirms that the observed features are highly predictive of the acceptance decision, constituting strong evidence of MAR violation. In the missing-data framework of Little and Rubin (2019), MAR requires that the probability of being observed (accepted) depends only on observed variables and not on the unobserved outcome; the near-perfect classification of accepted versus rejected applicants from the shared features alone implies that the selection mechanism is substantially

Table 2 Hyperparameter search grids for each classifier.

Model	Parameter	Values
XGBoost	n_estimators	100, 200, 300
	max_depth	3, 5, 7
	learning_rate	0.01, 0.05, 0.1
	subsample	0.7, 0.8, 1.0
Logistic Regression	C	0.001–10.0 (9 values)
	penalty	ℓ_1, ℓ_2
LogisticCEMD2	λ	0.001–10.0 (11 values)
MLP	hidden_layers	(50,), (100,), (50,50)
	activation	ReLU, tanh
	alpha	0.0001, 0.001, 0.01
Random Forest	n_estimators	100, 200, 300
	max_depth	None, 10, 20
LightGBM	num_leaves	31, 50, 70
	learning_rate	0.01, 0.05, 0.1
SVM	C	0.1, 1, 10, 100
	kernel	linear, RBF, polynomial

Table 3 MAR validation: bootstrap logistic regression classifying accepted versus rejected applications ($n = 500$ iterations).

Metric	Value
AUC (mean $\pm$ SD)	0.943 ± 0.001
AUC (95% CI)	[0.940, 0.945]
Hosmer-Lemeshow χ^2	647.6 ($p < 0.001$)

non-ignorable. We therefore interpret all subsequent CEM results as characterizing algorithm behavior under MNAR conditions. The documented MAR violation means that CEM's implicit assumption of ignorable selection is substantially violated, and all semi-supervised performance metrics should be interpreted in this light. This framing (studying CEM under documented assumption violation rather than assuming the violation away) allows us to characterize the conditions under which the method remains practically useful despite theoretical misspecification.

3.2 Regularization Tuning

Five-fold cross-validation across 11 λ values identified $\lambda = 5.0$ as optimal, yielding a validation AUC of 0.607 with 33.6 selected features out of 44 polynomial terms (Table 4). The finer grid reveals a gradual transition: validation AUC remains flat at approximately 0.582 for $\lambda \leq 1.0$, rises through 0.596 at $\lambda = 3.0$ and 0.599 at $\lambda = 4.0$, peaks at 0.607 at $\lambda = 5.0$, then stabilizes near 0.604 for $\lambda \geq 7.0$. The overfit metric (training AUC $-$ validation AUC) decreased monotonically from 0.057 at $\lambda = 0.001$ to 0.031 at $\lambda = 5.0$. Stronger regularization ($\lambda = 10.0$) reduced the

Table 4 Regularization tuning: 5-fold cross-validation results for LogisticCEMD2 across λ values.

λ	Train AUC	Val AUC	Features	Overfit
0.001	0.639	0.582	41.0	0.057
0.01	0.640	0.581	41.0	0.059
0.1	0.639	0.584	40.8	0.056
0.5	0.639	0.582	38.8	0.057
1.0	0.639	0.583	37.8	0.056
2.0	0.639	0.585	36.6	0.054
3.0	0.639	0.596	35.6	0.042
4.0	0.637	0.599	34.8	0.039
5.0	**0.638**	**0.607**	**33.6**	**0.031**
7.0	0.637	0.605	32.6	0.033
10.0	0.636	0.604	29.2	0.032

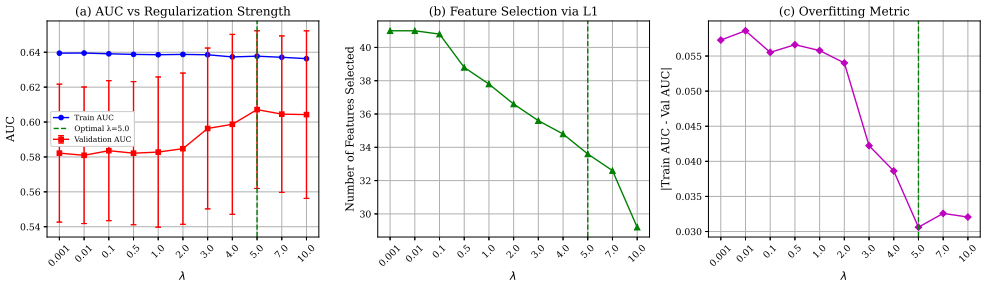


Fig. 1 Regularization tuning results for LogisticCEMD2 across 11 λ values. Left: validation AUC versus λ showing a gradual increase peaking at $\lambda = 5.0$. Center: number of selected features (out of 44 polynomial terms) decreasing monotonically with regularization strength. Right: overfit metric (training AUC − validation AUC) minimized at $\lambda = 5.0$.

number of features to 29.2 with minimal validation AUC loss (0.604), confirming that ℓ_1 regularization effectively controls model complexity. Training AUC remained stable across all λ values (0.636–0.640), indicating that the bias-variance tradeoff primarily affects generalization.

Figure 1 visualizes the relationship between regularization strength and model performance. The three panels show that validation AUC exhibits a sharp increase at $\lambda = 5.0$, the number of selected features decreases monotonically with increasing λ, and the overfit metric (training − validation AUC) reaches its minimum at the optimal regularization level.

3.3 Initialization Analysis

Among five initialization strategies evaluated across 15 repeats each, Semi-random and Baseline achieved the highest mean AUC values (0.606 ± 0.024 and 0.605 ± 0.024, respectively), with both strategies typically reaching the maximum iteration limit (Table 5). Random initialization converged faster (26.7 ± 11.6 iterations) but to inferior solutions (AUC $= 0.527 \pm 0.035$). K-means produced intermediate results

Table 5 Initialization strategy comparison (15 repeats per strategy, mean ± std).

Strategy	AUC	Final LL	Iterations	Time (s)
Semi-random	0.606±.024	−2,867	49.2±3.0	61.4
Baseline	0.605±.024	−2,864	46.1±8.5	68.0
K-means	0.570±.024	−2,591	40.5±10.0	90.6
Random	0.527±.035	−6,058	26.7±11.6	1.1
Conservative	0.525±.018	−2,655	49.7±1.0	62.3

(AUC = 0.570 ± 0.024), while Conservative initialization performed worst (AUC = 0.525 ± 0.018) despite achieving the lowest final log-likelihood (−2,655), suggesting convergence to a local optimum that assigns most unlabeled data to one class. The variability across initializations (mean AUC range: 0.525–0.606) indicates sensitivity to starting conditions, consistent with the non-convexity of the CEM objective (see also Figure B3 in Appendix B).

3.4 Model Comparison

Table 6 presents the mean performance across 25 runs for all 12 classifiers. The Friedman test rejected the null hypothesis of equal performance for all metrics: AUC ($\chi^2 = 193.4$, $p < 0.001$), accuracy ($\chi^2 = 148.0$, $p < 0.001$), F1 ($\chi^2 = 175.2$, $p < 0.001$), and recall ($\chi^2 = 162.4$, $p < 0.001$). The Nemenyi critical difference at $\alpha = 0.05$ was CD = 3.33.

Logistic Regression achieved the highest AUC (0.633, mean rank 1.60), followed by SuperLearner (0.628, rank 2.40) and SVM (0.620, rank 3.92). LogisticCEMD2 ranked sixth in AUC (0.602, rank 6.64) but fourth in F1 (0.608, rank 4.84) and fifth in recall (0.656, rank 4.98). The linear LogisticCEM achieved the highest recall (0.830, rank 1.08) but the lowest precision (0.534, rank 11.20), indicating aggressive classification toward the default class.

Among semi-supervised methods, LogisticCEMD2 outperformed LogisticCEM in AUC by 0.033 and outperformed LabelPropagation by 0.048. PseudoLabel-XGB (AUC 0.601) performed comparably to LogisticCEMD2, while CEM-XGBoost (0.593) and LabelPropagation (0.554) performed worse.

Brier scores (Table 6) reveal a calibration hierarchy distinct from the discrimination ranking. Logistic Regression achieved the lowest Brier score (0.236), closely followed by SuperLearner (0.237) and SVM (0.239). The CEM-based methods exhibited substantially worse calibration: LogisticCEMD2 (0.304) and CEM-XGBoost (0.303) scored over 28% higher (worse) than the supervised baseline, while LogisticCEM (0.325) exhibited the worst calibration among all 12 classifiers. This calibration gap is a consequence of the hard C-step, which assigns deterministic pseudo-labels and distorts the predicted probability distribution; the ablation study (Section 3.6) confirms that replacing hard assignment with soft posterior weights eliminates this miscalibration. Figure 2 presents the distribution of AUC scores across all 25 runs.

Table 6 Model comparison across 25 independent runs ($N_A = 2{,}000$, $N_R = 25{,}000$). Models ranked by mean AUC. Brier score measures calibration (lower is better). Friedman test: $\chi^2 = 193.4$, $p < 0.001$; Nemenyi CD = 3.33.

Model	AUC	Rank	F1	Recall	Precision	Brier
Logistic Regression	0.633±.025	1.60	0.612±.024	0.634±.034	0.592±.021	0.236±.005
SuperLearner	0.628±.024	2.40	0.606±.023	0.624±.032	0.589±.024	0.237±.005
SVM	0.620±.027	3.92	0.601±.020	0.621±.029	0.584±.029	0.239±.005
Random Forest	0.615±.025	4.44	0.588±.019	0.597±.026	0.580±.024	0.241±.006
MLP	0.605±.027	6.28	0.583±.027	0.596±.040	0.571±.025	0.252±.008
LogisticCEMD2	**0.602**±.027	**6.64**	**0.608**±.032	**0.656**±.080	0.573±.030	0.304±.013
XGBoost	0.602±.025	6.80	0.576±.018	0.583±.023	0.569±.021	0.252±.008
PseudoLabel-XGB	0.601±.024	6.96	0.575±.019	0.582±.027	0.569±.020	0.256±.007
CEM-XGBoost	0.593±.022	8.04	0.603±.034	0.645±.077	0.570±.026	0.303±.012
LightGBM	0.586±.024	9.32	0.565±.021	0.569±.029	0.561±.019	0.267±.010
LogisticCEM	0.569±.029	10.04	0.648±.022	0.830±.072	0.534±.015	0.325±.010
LabelPropagation	0.554±.025	11.56	0.550±.019	0.559±.026	0.542±.019	0.279±.011

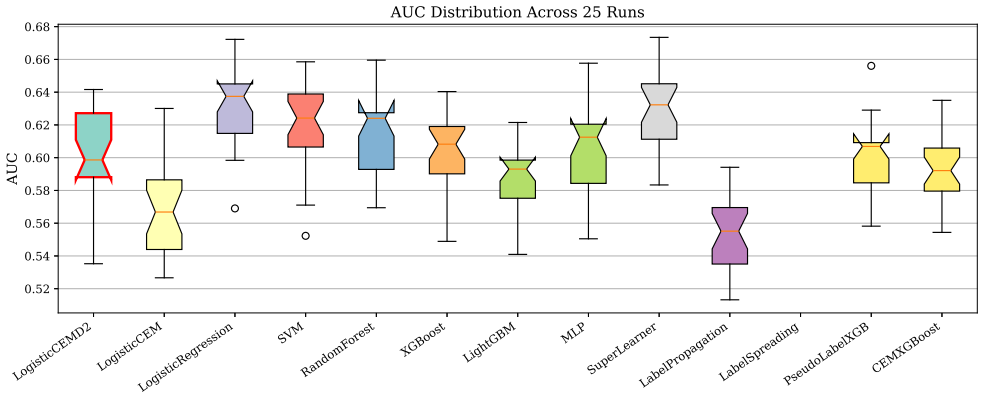


Fig. 2 Distribution of AUC scores across 25 independent runs for all 12 classifiers. Box plots show median, interquartile range, and outliers. Logistic Regression achieves the highest median AUC, while LogisticCEMD2 shows moderate variability consistent with the stochastic nature of the CEM procedure.

3.5 Large-Sample Model Comparison

To assess whether the classifier ranking changes at larger sample sizes, we repeated the model comparison at $N_A = 50{,}000$ with $N_R = 25{,}000$ across 10 independent runs. SVM was replaced with LinearSVC (wrapped in `CalibratedClassifierCV`) due to the $\mathcal{O}(n^2)$ complexity of kernel SVM at this scale. Table 7 presents the results.

At $N_A = 50{,}000$, the classifier ranking shifts substantially compared to the small-sample setting ($N_A = 2{,}000$). Ensemble methods (SuperLearner, AUC = 0.646; XGBoost, 0.644; RandomForest, 0.643) now occupy the top positions, while LogisticCEMD2 (0.620) and LogisticCEM (0.608) fall to 9th and 10th place, respectively. This contrasts with the small-sample results (Table 6) where LogisticCEMD2 (AUC = 0.602) ranked 6th of 12 classifiers. The AUC gap between the top-ranked model

Table 7 Large-sample model comparison across 10 independent runs ($N_A = 50{,}000$, $N_R = 25{,}000$). Models ranked by mean AUC. Brier score measures calibration (lower is better).

Model	AUC	F1	Recall	Precision	Brier
SuperLearner	0.646±.004	0.611±.005	0.623±.007	0.600±.005	0.233±.001
XGBoost	0.644±.004	0.609±.004	0.619±.007	0.599±.004	0.234±.001
RandomForest	0.643±.004	0.609±.005	0.622±.009	0.597±.005	0.234±.001
MLP	0.643±.004	0.608±.016	0.621±.039	0.597±.007	0.234±.001
LightGBM	0.641±.004	0.607±.004	0.617±.007	0.597±.004	0.235±.001
LogReg	0.639±.005	0.611±.005	0.630±.006	0.593±.005	0.235±.001
LinearSVC	0.639±.005	0.611±.005	0.630±.006	0.592±.005	0.235±.001
CEM-XGBoost	0.635±.005	0.619±.006	0.650±.013	0.592±.005	0.240±.001
LogisticCEMD2	0.620±.004	0.626±.003	0.688±.005	0.575±.005	0.248±.002
LogisticCEM	0.608±.005	0.627±.004	0.708±.005	0.563±.004	0.256±.002

and LogisticCEMD2 narrowed from 0.031 at $N_A = 2{,}000$ (Logistic Regression: 0.633) to 0.026 at $N_A = 50{,}000$ (SuperLearner: 0.646), while absolute AUC levels improved for all models. The calibration hierarchy persists at the larger sample size: supervised methods achieve Brier scores of 0.233–0.235, while the CEM variants exhibit worse calibration (LogisticCEMD2: 0.248, LogisticCEM: 0.256), consistent with the hard C-step miscalibration documented in the ablation study. CEM-XGBoost (0.240) occupies an intermediate position, suggesting that the tree-based component partially compensates for pseudo-label noise. The CEM variants retain the highest recall values (LogisticCEM: 0.708, LogisticCEMD2: 0.688) and F1 scores, confirming that the semi-supervised component consistently shifts the decision boundary toward higher sensitivity at the expense of precision. This recall advantage persists even as discrimination (AUC) falls behind ensemble methods.

3.6 Ablation Study

The ablation study (Table 8) reveals that individual components contribute asymmetrically to performance. Adding the semi-supervised CEM component alone with hard label assignment (Base+SSL) reduced AUC from 0.633 to 0.618 (Wilcoxon $W = 20.0$, Bonferroni-corrected $p < 0.001$), confirming that hard pseudo-label propagation without regularization degrades discrimination. Polynomial features without SSL (Base+Poly) maintained AUC at 0.628, while adding ℓ_1 regularization (Base+Poly+ℓ_1) yielded 0.630 with improved precision.

The FullModel with hard C-step (AUC = 0.606) outperformed Base+SSL+Poly (0.592; $W = 33.0$, Bonferroni-corrected $p = 0.001$) but fell below both the CEM-only model (Base+SSL, 0.618) and the supervised Base (0.633). SelfTraining achieved AUC 0.623 and significantly outperformed the FullModel ($W = 25.0$, Bonferroni-corrected $p < 0.001$).

The soft EM variants, which retain fractional posterior weights instead of binarizing at 0.5, produced the most informative result: SoftEM (AUC = 0.633) matched the supervised Base and significantly outperformed the hard FullModel ($W = 1.0$, Bonferroni-corrected $p < 0.001$). SoftEM+Linear (AUC = 0.633) similarly outperformed its hard counterpart Base+SSL ($W = 19.0$, Bonferroni-corrected $p < 0.001$).

Table 8 Ablation study results (25 independent runs). Brier score measures calibration (lower is better). Wilcoxon signed-rank tests with Bonferroni correction ($\alpha_{\text{adj}} = 0.0083$) compare six pre-specified pairs.

Configuration	AUC	Accuracy	F1	Recall	KS	Brier
Base (LogReg)	0.633±.025	0.598±.023	0.612±.024	0.634±.034	0.220±.042	0.236±.005
Base+SSL	0.618±.033	0.585±.025	0.627±.019	0.699±.037	0.200±.048	0.320±.014
Base+Poly	0.628±.023	0.592±.022	0.603±.021	0.618±.029	0.214±.041	0.239±.005
Base+Poly+ℓ_1	0.630±.023	0.593±.023	0.612±.023	0.641±.036	0.217±.039	0.237±.005
Base+SSL+Poly	0.592±.022	0.571±.019	0.586±.018	0.606±.027	0.169±.038	0.321±.016
FullModel (hard)	0.606±.026	0.581±.027	0.598±.028	0.625±.050	0.190±.046	0.302±.013
SelfTraining	0.623±.029	0.587±.023	0.625±.019	0.688±.029	0.203±.042	0.314±.013
SoftEM	**0.633**±.024	**0.596**±.024	0.608±.025	0.629±.037	**0.221**±.045	**0.236**±.005
SoftEM+Linear	0.633±.025	0.598±.023	0.612±.024	0.634±.034	0.219±.042	0.236±.005
Wilcoxon signed-rank tests (AUC, Bonferroni-corrected):						
Base vs. Base+SSL	$W = 20.0$, $p_{\text{corr}} < 0.001$					
Base+SSL vs. FullModel	$W = 71.0$, $p_{\text{corr}} = 0.075$ (n.s.)					
Base+SSL+Poly vs. FullModel	$W = 33.0$, $p_{\text{corr}} = 0.001$					
SelfTraining vs. FullModel	$W = 25.0$, $p_{\text{corr}} < 0.001$					
FullModel vs. SoftEM	$W = 1.0$, $p_{\text{corr}} < 0.001$					
Base+SSL vs. SoftEM+Linear	$W = 19.0$, $p_{\text{corr}} < 0.001$					

The Brier score differences are particularly revealing: SoftEM achieved 0.236, matching the supervised Base (0.236), while the hard FullModel scored 0.302. On a representative split, the Hosmer-Lemeshow goodness-of-fit test failed to reject calibration for SoftEM ($\chi^2 = 13.94$, $p = 0.083$) but rejected it decisively for the hard FullModel ($\chi^2 = 4{,}365$, $p < 0.001$). Figure 5 presents the calibration curves (reliability diagrams). These results demonstrate that the discrimination loss previously attributed to the semi-supervised component was caused by the hard C-step binarization, not by the CEM framework itself. When posteriors are retained as continuous weights, the semi-supervised model recovers both discrimination and calibration.

Figure B4 in Appendix B visualizes the ablation results across all nine configurations.

3.7 Reject Size Sensitivity

In Scenario 1 ($N_A = 2{,}000$), AUC showed a slight declining trend across unlabeled sample sizes, from 0.635 ± 0.024 ($N_R = 1{,}000$) to 0.620 ± 0.020 ($N_R = 25{,}000$) across 5 repeats per condition (Table 9, Figure 3). The decline is modest relative to the within-condition variability, indicating that the model tolerates moderate volumes of unlabeled data without substantial degradation.

In Scenario 2 (Table 9, Figure 4), where the labeled sample was expanded to $N_A \approx 494{,}852$ (after undersampling the majority class to balance), AUC remained stable at approximately 0.645 for $N_R \leq 10{,}000$, then declined to 0.631 ± 0.001 at $N_R = 100{,}000$ and 0.602 ± 0.003 at $N_R = 1{,}000{,}000$ (Table 9). At $N_R = 2{,}064{,}314$ (the full rejected dataset), AUC recovered slightly to 0.607 ± 0.001, suggesting a partial stabilization when all available unlabeled data is included. The sharpest decline occurred when N_R exceeded N_A by approximately a factor of two. The small standard deviations in Scenario 2 (≤ 0.003) relative to Scenario 1 (≥ 0.017) reflect the much larger labeled sample providing a more stable training anchor. The overall degradation pattern is

Table 9 Reject size sensitivity: AUC of LogisticCEMD2 as a function of unlabeled sample size N_R (mean ± std over 5 repeats).

N_R	**Scenario 1** ($N_A = 2{,}000$)	**Scenario 2** ($N_A \approx 494{,}852$)
1,000	0.635 ± 0.024	0.645 ± 0.001
2,000	0.630 ± 0.024	–
5,000	0.625 ± 0.027	–
10,000	0.623 ± 0.026	0.644 ± 0.001
15,000	0.622 ± 0.017	–
20,000	0.621 ± 0.020	–
25,000	0.620 ± 0.020	–
100,000	–	0.631 ± 0.001
1,000,000	–	0.602 ± 0.003
2,064,314	–	0.607 ± 0.001

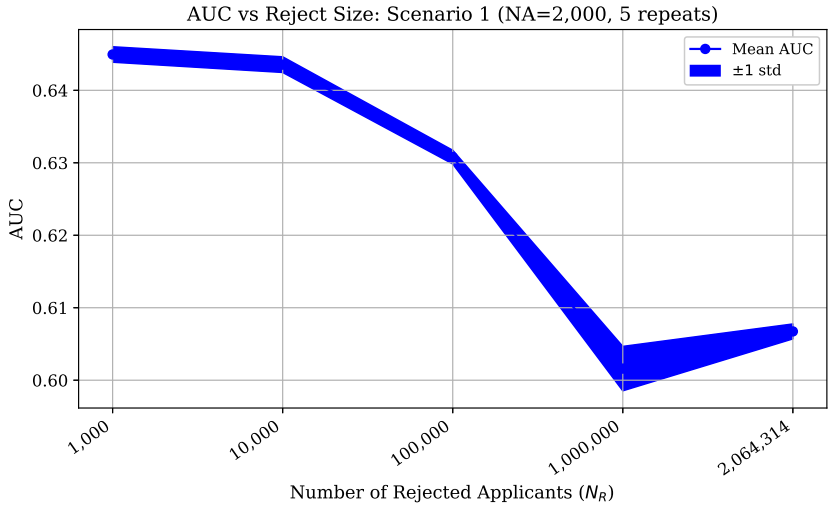


Fig. 3 Scenario 1: AUC of LogisticCEMD2 as a function of unlabeled sample size N_R with fixed $N_A = 2{,}000$. AUC exhibits a slight declining trend from 0.635 ($N_R = 1{,}000$) to 0.620 ($N_R = 25{,}000$), though within-condition variability (error bands) substantially overlaps across all settings.

consistent with label noise accumulation: as more unlabeled samples receive pseudo-labels through the CEM process, errors in the C-step propagate through subsequent EM iterations, progressively biasing the model.

4 Discussion

The central finding of this study is that the LendingClub selection mechanism is strongly non-random, with near-perfect classification of accepted versus rejected applicants from the shared features alone, establishing that the Missing at Random assumption underlying CEM-based reject inference is substantially violated. All subsequent results must therefore be interpreted as characterizing CEM behavior under MNAR conditions rather than validating CEM as a principled correction

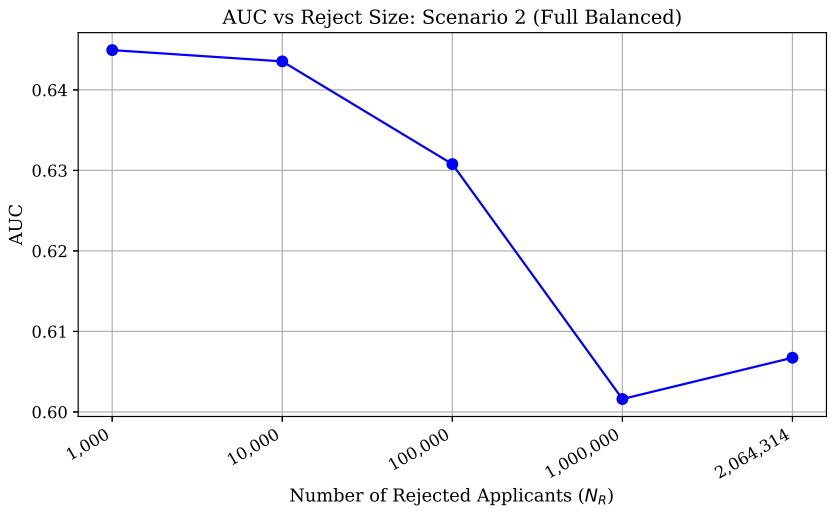


Fig. 4 Scenario 2: AUC of LogisticCEMD2 with full balanced labeled dataset ($N_A \approx 494{,}852$ after undersampling). AUC declines as N_R increases, with the sharpest drop when N_R exceeds N_A by a factor of two, and a slight recovery at the full rejected population.

for selection bias. Under these conditions, supervised methods consistently outperformed semi-supervised approaches in overall discrimination across 25 independent runs, while LogisticCEMD2 exhibited a distinctive advantage in default detection sensitivity. The ablation study revealed an asymmetric contribution pattern: polynomial feature expansion and ℓ_1 regularization preserved classification performance near the supervised baseline, whereas the semi-supervised CEM component alone reduced discrimination while substantially increasing recall.

These findings contribute to a longstanding question in credit scoring: whether incorporating rejected applicants improves predictive models. Lessmann et al. (2015) benchmarked 41 classifiers across multiple credit datasets and concluded that no single algorithm dominates universally. Our results corroborate this conclusion in the specific context of P2P lending with reject inference, while extending the comparison to include semi-supervised methods evaluated with formal Friedman and Nemenyi statistical tests against supervised baselines on a dataset of over three million applications. Baesens et al. (2003) similarly documented the competitive performance of logistic regression across diverse credit scoring contexts, a pattern that persists here despite the availability of more flexible alternatives such as gradient boosting and neural networks. Anderson and Hardin (2013) demonstrated that linear logistic CEM improved credit scoring on simulated data where the labeled-to-unlabeled ratio was moderate. Our analysis extends their work to real-world LendingClub data two orders of magnitude larger and reveals a boundary condition their simulations could not capture: at an extreme acceptance rate (8%), the linear CEM achieves the highest recall among all classifiers but at the cost of the lowest precision, producing an aggressive default classifier. The polynomial extension partially corrects this recall-precision imbalance

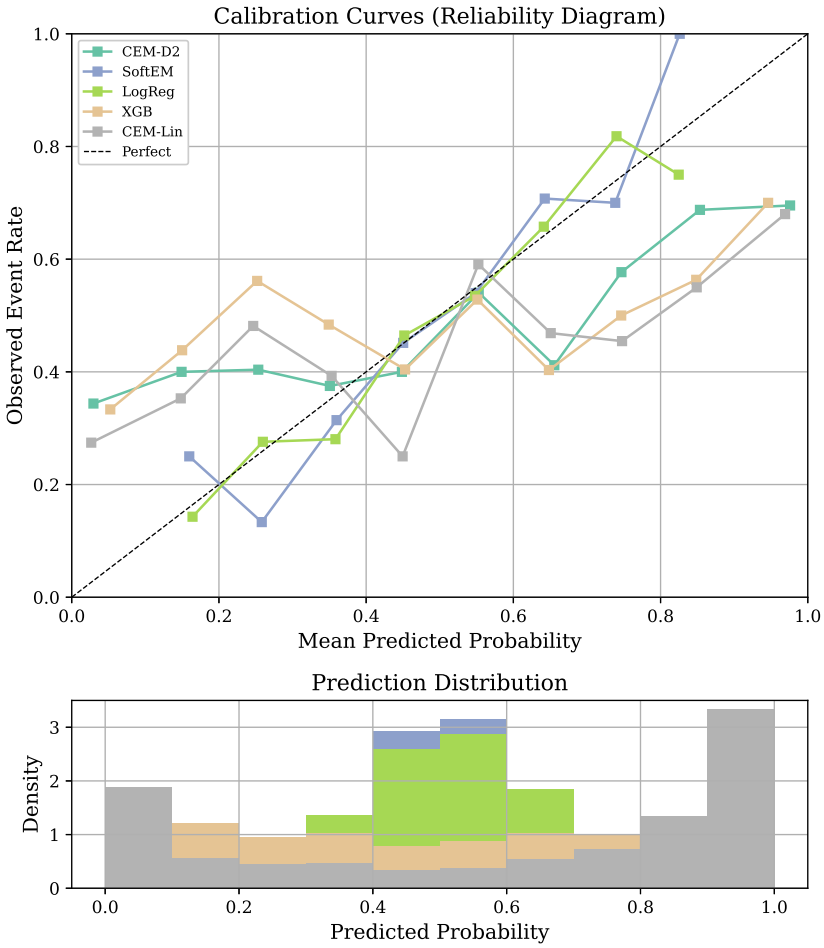


Fig. 5 Calibration curves (reliability diagrams) for key models on a representative split. SoftEM and Logistic Regression track the perfect calibration diagonal, whereas the hard LogisticCEMD2 and LogisticCEM exhibit systematic miscalibration from the deterministic pseudo-label assignment. The lower panel shows the distribution of predicted probabilities.

by capturing pairwise feature interactions (approximately three-quarters of the 44 polynomial features survived ℓ_1 selection) that a linear boundary cannot represent.

That supervised logistic regression achieved the highest discrimination among classifiers using the hard C-step warrants examination, because it contradicts expectations from the semi-supervised learning literature. Li et al. (2017) demonstrated semi-supervised SVM gains for reject inference on P2P lending data, and the general SSL framework predicts improvement when unlabeled data reveals cluster structure aligned with class boundaries (Chapelle et al. 2006). Our ablation results identify a specific mechanism for the divergence: the hard C-step assigns deterministic pseudo-labels by binarizing posterior probabilities at 0.5. When the model is trained on a small, potentially unrepresentative labeled sample drawn from a strongly non-random

selection process, early pseudo-labels may reinforce rather than correct the initial model's biases. Each subsequent EM iteration compounds these errors, progressively degrading both discrimination and calibration.

The soft EM comparison (Table 8) isolates this mechanism empirically. Replacing the hard C-step with fractional posterior weights eliminates the discrimination penalty entirely: SoftEM matches the supervised baseline in both AUC (0.633) and Brier score (0.236), while the hard FullModel achieves only 0.606 and 0.302, respectively. The Hosmer-Lemeshow test confirms the calibration difference: SoftEM passes ($\chi^2 = 13.94$, $p = 0.083$) while the hard variant fails ($\chi^2 = 4{,}365$, $p < 0.001$). A natural question is whether SoftEM achieves this match by effectively neutralizing the unlabeled contribution, recovering the supervised solution rather than incorporating information from rejected applicants. To investigate, we compared the fitted coefficient vectors of Base and SoftEM on a representative run (seed = 42). The two vectors are highly similar: Pearson correlation $r = 0.950$, cosine similarity 0.945, and ℓ_2 distance 0.18. By contrast, the hard FullModel diverges from the supervised baseline ($r = 0.137$, cosine similarity 0.132, ℓ_2 distance 4.33). SoftEM retained 13 non-zero coefficients (of 44 polynomial features) compared to 20 for Base and 30 for the hard variant. These results confirm that under strong MAR violation, the soft posterior weights downweight the unlabeled observations to near-zero effective influence, and the SoftEM solution converges to the supervised fit. SoftEM "works" not because it successfully extracts information from rejected applicants, but because it avoids contaminating the supervised signal. This is a weaker claim than successful semi-supervised learning, yet a practically useful one: SoftEM provides a safe default that preserves discrimination and calibration regardless of whether the unlabeled data distribution is informative. This finding aligns with the classical EM literature (Dempster et al. 1977), where the E-step retains the full posterior distribution precisely to avoid the information loss that hard assignment introduces. El Annas et al. (2023), who applied Hidden Markov Models for semi-supervised credit scoring on LendingClub data, reported discrimination values for semi-supervised approaches in a comparable range, suggesting that hard label assignment limitations extend beyond CEM to other semi-supervised frameworks under strong MAR violation.

Sensitivity analyses yielded two additional findings with practical relevance. Under small labeled samples, LogisticCEMD2 exhibited a slight declining trend in AUC (0.635 to 0.620) across a 25-fold range of unlabeled data volumes, though within-condition variability was large relative to this decline, indicating that moderate amounts of rejected applicant data can be incorporated without substantial degradation. Under large labeled samples, however, discrimination declined by more than four AUC percentage points as the unlabeled sample grew from one thousand to one million observations, with a partial recovery when the full rejected population was included. This pattern aligns with theoretical expectations: as the number of pseudo-labeled observations grows relative to the labeled anchor, the CEM objective increasingly reflects the pseudo-label distribution rather than the true outcome distribution. The initialization analysis confirmed the non-convexity of the CEM objective function, with AUC ranging from 0.525 to 0.606 across five initialization strategies. Supervised warm-starting consistently reached higher-quality solutions than data-driven alternatives

such as K-means clustering, consistent with Celeux and Govaert's (1992) observation that CEM behavior depends critically on the initial partition.

The relationship between CEM and econometric approaches to sample selection deserves comment. Heckman's (1979) two-stage correction is the classical econometric solution: a probit selection equation estimates the probability of acceptance, and the inverse Mills ratio from this equation is included as a covariate in the outcome equation to correct for selection bias. However, Heckman correction requires an *exclusion restriction*, at least one variable that affects the selection decision but not the outcome (Puhani 2000). In credit scoring, this restriction is rarely available because the same features (FICO score, DTI, loan amount) simultaneously drive both acceptance decisions and default outcomes (Banasik et al. 2003). CEM avoids the exclusion restriction by treating rejected applicants as unlabeled rather than as a selection-corrected subsample, but introduces the MAR assumption instead. Our results demonstrate that under strong MAR violation, CEM's practical contribution is limited to recall improvement rather than bias correction, a finding consistent with Manski's (1989) observation that without strong assumptions about the selection process, identification of the outcome distribution for the unselected population is fundamentally limited.

To illustrate the economic implications, consider a portfolio of 1,000 applications at the observed 20% default rate. The hard LogisticCEMD2's recall advantage over logistic regression translates to approximately 4 additional detected defaults per 1,000 applications. At the mean loan amount in the dataset ($14,600; Table 1) and assuming 85% loss-given-default, a standard estimate for unsecured consumer loans (Thomas 2000), each missed default costs approximately $12,400, yielding a potential loss reduction of roughly $50,000 per 1,000 applications. However, we note that Expected Maximum Profit (EMP) (Verbraken et al. 2014) values are negative across all models (Table 6), indicating that at this sample size and feature set, no classifier achieves positive economic profit under the Verbraken cost framework. This reflects the constrained feature set (8 variables) and the strong MAR violation rather than a deficiency of any individual model. The SoftEM variant provides a stronger economic proposition: it matches the supervised baseline in discrimination while retaining the CEM framework's ability to incorporate rejected applicant information, offering a path toward improved profit once richer feature sets become available for both accepted and rejected populations. The relative economic value depends on the institution's cost structure: when the cost of undetected defaults substantially exceeds the cost of false declines, the CEM framework's sensitivity advantage may compensate, particularly when using soft rather than hard label assignment.

For credit risk practitioners, these results suggest two recommendations. First, the soft EM variant should be preferred over the hard C-step: it matches supervised discrimination and calibration while retaining the ability to incorporate unlabeled rejected applicant data, and requires no additional tuning parameters. Second, the hard C-step variant remains useful when false negatives (undetected defaults) carry substantially higher costs than false positives (denied creditworthy borrowers), as it shifts the precision-recall trade-off toward sensitivity (Verbraken et al. 2014; Hand 2009). In both cases, practitioners should warm-start the algorithm from a supervised fit, control the unlabeled-to-labeled ratio below approximately 50:1, and select

the regularization penalty via cross-validation with unlabeled data properly passed to the training procedure. The polynomial features and ℓ_1 regularization improve performance regardless of the label assignment strategy, making them independently valuable for credit scoring with interaction effects.

This study has limitations that affect interpretation. The feature set is restricted to eight variables common to both datasets, discarding 142 features available for accepted loans only; this constraint is inherent to reject inference and systematically underestimates all models' discriminative potential, meaning that absolute performance values reported here are lower bounds; however, because the feature restriction affects all classifiers equally, relative comparisons and ranking conclusions remain valid even if absolute performance would improve with richer feature sets. The dataset spans 2007–2018, encompassing the financial crisis, Dodd-Frank Act, and CFPB establishment. Temporal non-stationarity in default rates and acceptance criteria likely biases our evaluation toward underestimating within-regime performance, because pooling across heterogeneous periods introduces distributional shift between training and test data. We use random rather than temporal splitting because the year variable is unavailable for rejected applicants, precluding temporal stratification that would maintain the reject inference framework. This design ensures each fold contains observations from across the full time span, though it does not simulate prospective deployment. A temporal split (training on pre-2015 data, testing on 2015–2018) would better evaluate real-world deployment but would require either discarding reject inference entirely or imputing temporal information for rejected applicants; this remains an important direction for future work. Generalizability to other platforms or macroeconomic conditions is untested; LendingClub's extremely low acceptance rate (8%) likely produces stronger MAR violations than platforms with more permissive lending policies, which would bias our results toward underestimating CEM's utility in settings where selection is less extreme. The algorithmic framework itself makes no platform-specific assumptions, and LendingClub is the largest publicly available P2P lending dataset, making it the natural benchmark for reject inference methods. As documented in Section 3.1, the strongly non-random selection mechanism means that all CEM results reflect behavior under MNAR conditions; the semi-supervised component's contribution would likely differ in settings where MAR approximately holds. The labeled samples in the model comparison and ablation experiments ($N_A = 2{,}000$) are smaller than what would be available in production. Our large-sample replication (Section 3.5, $N_A = 50{,}000$) confirmed that the classifier ranking indeed shifts: ensemble methods (SuperLearner, XGBoost, RandomForest) rose to the top positions, while LogisticCEMD2 fell to 9th of 10 models by AUC. However, the CEM variants retained the highest recall values (LogisticCEM: 0.708, LogisticCEMD2: 0.688), indicating that the semi-supervised recall advantage persists even when overall discrimination favors ensemble methods.

LogisticCEMD2 extends the Classification EM framework with polynomial features and ℓ_1 regularization for reject inference in credit scoring. On LendingClub data where the selection mechanism is strongly non-random, the hard C-step variant trades discrimination for sensitivity, while the soft EM variant preserves both discrimination and calibration at the level of the supervised baseline. The soft assignment approach

eliminates the primary limitation of the original CEM formulation and should be preferred in practice. Evaluation on additional lending platforms, under alternative semi-supervised paradigms such as consistency regularization or co-training, with explicit Missing Not at Random correction methods, and under temporal validation protocols would clarify the conditions under which reject inference reliably improves credit scoring beyond supervised approaches.

Data Availability

The LendingClub dataset is publicly available on Kaggle (https://www.kaggle.com/wordsforthewise/lending-club). The complete reproducibility package, including the LogisticCEMD2 implementation, all baseline model configurations, the full experimental pipeline (preprocessing, splitting, evaluation), hyperparameter grids, random seeds, and scripts to reproduce every table and figure, is available at https://github.com/isamil/polynomial-cem-credit-scoring.

Funding

This research received no external funding.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Abderrahim El Amrani: Methodology, Formal Analysis, Writing (Original Draft), Visualization. Badreddine Benyacoub: Conceptualization, Supervision, Resources, Writing (Review and Editing). Mohammed El Haj Tirari: Conceptualization, Supervision, Writing (Review and Editing), Project Administration. Ismail El Moudden: Conceptualization, Methodology, Supervision, Formal Analysis, Writing (Review and Editing), Visualization. All authors read and approved the final manuscript.

References

Amini, M.-R., Gallinari, P.: Semi-supervised logistic regression. In: Proceedings of the 15th European Conference on Artificial Intelligence (ECAI 2002), pp. 390–394. IOS Press, Lyon, France (2002)

Anderson, B., Hardin, J.M.: Modified logistic regression using the EM algorithm for reject inference. International Journal of Data Analysis Techniques and Strategies **5**(4), 359–373 (2013)

Bellotti, T., Crook, J.: Support vector machines for credit scoring and discovery of significant features. Expert Systems with Applications **36**(2), 3302–3308 (2009)

Banasik, J., Crook, J., Thomas, L.C.: Sample selection bias in credit scoring models. Journal of the Operational Research Society **54**(8), 822–832 (2003) https://doi.org/10.1057/palgrave.jors.2601578

Bücker, M., Kampen, M.v., Kraß, A.: Reject inference in consumer credit scoring with nonignorable missing data. Journal of Banking & Finance **37**(3), 1040–1045 (2013)

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J.: Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society **54**(6), 627–635 (2003)

Celeux, G., Govaert, G.: A classification EM algorithm for clustering and two stochastic versions. Computational Statistics & Data Analysis **14**(3), 315–332 (1992)

Chapelle, O., Schölkopf, B., Zien, A.: Semi-Supervised Learning. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA (2006)

Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research **7**, 1–30 (2006)

Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society: Series B (Methodological) **39**(1), 1–22 (1977)

El Annas, M., Benyacoub, B., Ouzineb, M.: Semi-supervised adapted HMMs for P2P credit scoring systems with reject inference. Computational Statistics **38**(1), 149–169 (2023) https://doi.org/10.1007/s00180-022-01220-9

Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research **9**, 1871–1874 (2008)

Feelders, A.: Credit scoring and reject inference with mixture models. International Journal of Intelligent Systems in Accounting, Finance & Management **9**(1), 1–8 (2000)

Hand, D.J.: Measuring classifier performance: a coherent alternative to the area under the roc curve. Machine Learning **77**(1), 103–123 (2009)

Heitjan, D.F., Basu, S.: Distinguishing "missing at random" and "missing completely at random". The American Statistician **50**(3), 207–213 (1996)

Heckman, J.J.: Sample selection bias as a specification error. Econometrica **47**(1), 153–161 (1979)

Hand, D.J., Henley, W.E.: Statistical classification methods in consumer credit scoring: a review. Journal of the Royal Statistical Society: Series A (Statistics in Society)

160(3), 523–541 (1997)

Lessmann, S., Baesens, B., Seow, H.-V., Thomas, L.C.: Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. European Journal of Operational Research **247**(1), 124–136 (2015)

LendingClub Corporation: LendingClub Corporation Annual Report 2024. https://www.lendingclub.com/company/annual-reports. Accessed: 2025-01-15 (2024)

Little, R.J.A., Rubin, D.B.: Statistical Analysis with Missing Data, 3rd edn. John Wiley & Sons, Hoboken, NJ (2019)

Li, Z., Tian, Y., Li, K., Zhou, F., Yang, W.: Reject inference in credit scoring using semi-supervised support vector machines. Expert Systems with Applications **74**, 105–114 (2017) https://doi.org/10.1016/j.eswa.2017.01.011

Manski, C.F.: Anatomy of the selection problem. Journal of Human Resources **24**(3), 343–360 (1989)

Murty, M.N., Raghava, R.: Kernel Methods and Support Vector Machines. Springer, Berlin (2016)

Puhani, P.A.: The Heckman correction for sample selection and its critique. Journal of Economic Surveys **14**(1), 53–68 (2000)

Shen, F., Zhao, X., Kou, G., Alsaadi, F.E.: A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique. Applied Soft Computing **98**, 106852 (2021) https://doi.org/10.1016/j.asoc.2020.106852

Thomas, L.C.: A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. International Journal of Forecasting **16**(2), 149–172 (2000)

Vittaut, J.-N., Amini, M.-R., Gallinari, P.: Learning classification with both labeled and unlabeled data. In: Machine Learning: ECML 2002. LNAI, vol. 2430, pp. 468–479. Springer, Berlin, Heidelberg (2002)

Verthez, W.: Lending Club Loan Data. https://www.kaggle.com/wordsforthewise/lending-club. Accessed: 2024-09-01 (2017)

Verbraken, T., Wouter, V., Baesens, B.: Development and application of consumer credit scoring models using profit-based classification measures. European Journal of Operational Research **238**(2), 505–513 (2014)

West, D., Dellana, S., Qian, J.: Neural network ensemble strategies for financial decision applications. Computers & Operations Research **32**(10), 2543–2559 (2005)

Wang, G., Hao, J., Ma, J., Jiang, H.: A comparative assessment of ensemble learning

for credit scoring. Expert Systems with Applications **38**(1), 223–230 (2011)

Wang, G., Ma, J., Huang, L., Xu, K.: Two credit scoring models based on dual strategy ensemble trees. Knowledge-Based Systems **26**, 61–68 (2012)

Zhu, X., Goldberg, A.B.: Introduction to Semi-Supervised Learning. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, San Rafael, CA (2009)

Appendix A Derivation of the Discriminative Log-Likelihood

Starting from the complete-data log-likelihood in the mixture model framework:

$$LL_c = \sum_{i=1}^{n} \sum_{m=1}^{M} z_{im} \log \left[\pi_m p(\mathbf{x}_i \mid \boldsymbol{\theta}_m) \right].$$

Applying Bayes' theorem: $\pi_m p(\mathbf{x}_i \mid \boldsymbol{\theta}_m) = P(P_m \mid \mathbf{x}_i) \cdot p(\mathbf{x}_i)$. Substituting:

$$LL_c = \sum_{i=1}^{n} \sum_{m=1}^{M} z_{im} \log P(P_m \mid \mathbf{x}_i) + \sum_{i=1}^{n} \log p(\mathbf{x}_i),$$

where the second term is constant with respect to $\boldsymbol{\beta}$ and can be ignored during optimization. The first term is the discriminative log-likelihood used in Eq. (1).

Appendix B Supplementary Tables and Figures

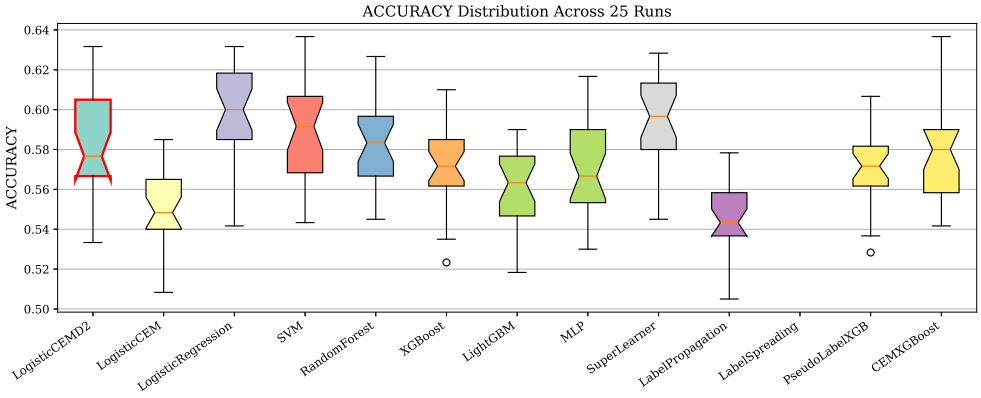


Fig. B1 Distribution of accuracy scores across 25 independent runs for all 12 classifiers. Friedman test: $\chi^2 = 148.0$, $p < 0.001$.

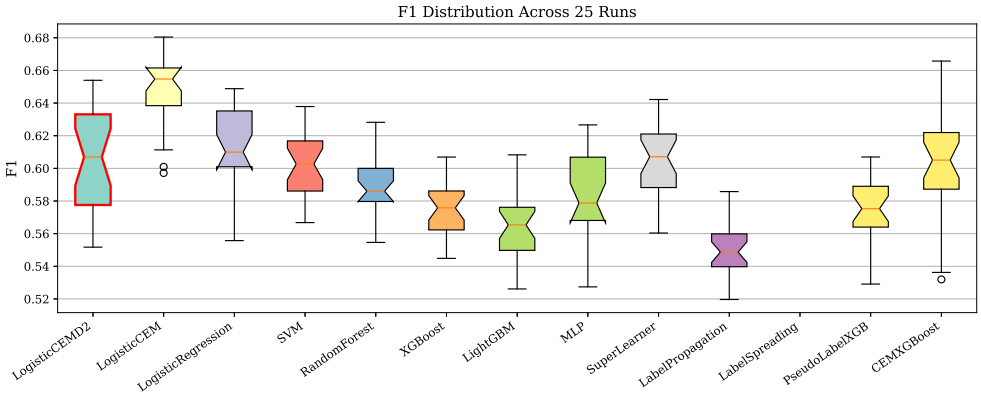


Fig. B2 Distribution of F1 scores across 25 independent runs for all 12 classifiers. LogisticCEM achieves the highest median F1 driven by high recall, while LogisticCEMD2 ranks fourth. Friedman test: $\chi^2 = 175.2$, $p < 0.001$.

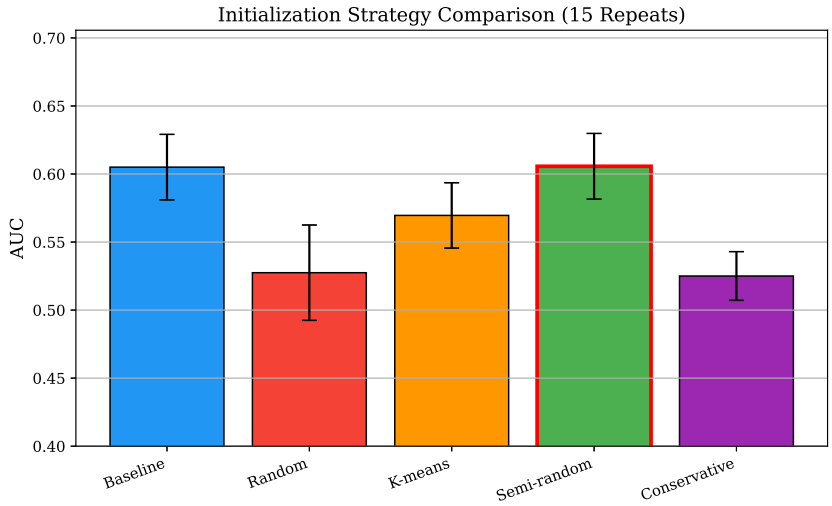


Fig. B3 Initialization strategy comparison showing AUC and final log-likelihood for five strategies across 15 repeats each. Semi-random and Baseline consistently achieve the highest AUC values, while Conservative initialization converges to poor local optima.

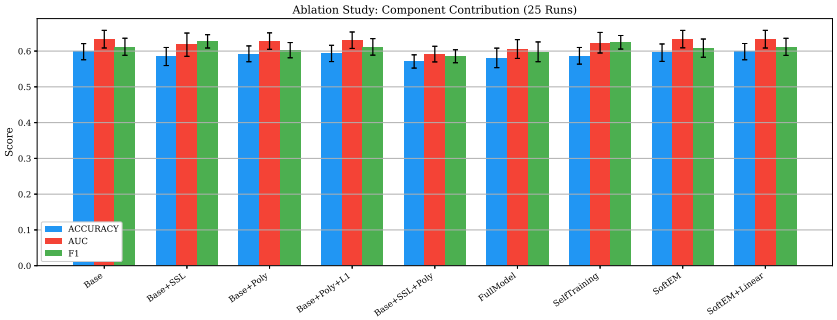


Fig. B4 Ablation study results across 25 independent runs for nine model configurations. The Base logistic regression and SoftEM variants achieve the highest AUC, while the hard SSL component alone (Base+SSL) reduces AUC but increases recall. The SoftEM variant recovers full discrimination by retaining fractional posterior weights instead of binarizing at 0.5.