Bridging Accuracy and Interpretability: A Rescaled Cluster-then-Predict Approach for Enhanced Credit Scoring

Huei-Wen Teng, Ming-Hsuan Kang, I-Han Lee, Le-Chi Bai

Abstract

Credit scoring is pivotal in the financial industry for assessing individuals' creditworthiness and optimizing financial institutions' risk-adjusted returns. While the XGBoost algorithm stands as the state-of-the-art classifier for credit scoring, its intricate nature impedes easy interpretation, a critical aspect for stakeholders' decision-making. This paper introduces a novel approach termed the "Rescaled Cluster-then-Predict Method," aimed at enhancing both the interpretability and predictive performance of credit scoring models. Our method employs a two-step process, initially rescaling the features and subsequently clustering the data into subgroups. Consequently, we employ Logistic Regression within each subgroup to generate predictions. The paper's primary contributions are twofold. Firstly, empirical evaluations on two distinct datasets demonstrate that our proposed method attains a competitive performance compared to XGBoost while substantially improving interpretability. Notably, in some instances, the Logistic Regression outperforms XGBoost. Secondly, we reveal that clustering solely the positive cases, as opposed to the entire dataset, yields comparable results while markedly reducing computational requirements. These insights hold significant practical implications for the financial industry, which consistently seeks credit scoring models that are not only accurate but also interpretable and computationally efficient.

Keywords: credit scoring, cluster-then-predict, rescaling, XGBoost, Logistic Regresion
JEL: C38, C53, G21

1. Introduction

A credit scoring model is a statistical tool used to predict the likelihood of a borrower defaulting on a loan. It assigns a score to the borrower based on their financial history and other factors, such as employment status, credit utilization, and debt-to-income ratio. Organizations that provide loans, credit products, or insurance policies use credit scoring models to assess the risk and make informed decisions (Thomas et al., 2017; Hand and Henley, 1997; Menard, 2002).

In recent years, novel machine learning algorithms have been compared with the traditional Logistic Regression for credit scoring. Ensemble methods, such as boosting, often outperform single classifiers (Baesens et al., 2003; Lessmann et al., 2015). Boosting combines weak models to form a stronger model, which can lead to better predictions for credit risk assessment. However, the complex ensemble nature of boosting can limit interpretability (Freund et al., 1999; Schapire, 1999; Friedman, 2001). Despite this limitation, boosting models are widely used for their predictive power in credit scoring applications and are considered a state-of-the-art method (Xia et al., 2017; Chang et al., 2018; Qiu, 2019; Qin et al., 2021; Liu et al., 2022).

In the context of credit scoring, where decisions have a real-life impact on loan approvals or insurance coverage, interpretable or explainable machine learning algorithms are crucial. Logistic Regression is favored for its interpretability, as the coefficients of independent variables reveal their relationship with the dependent variable, allowing stakeholders such as borrowers and regulators to understand the reasoning behind decisions. The importance of explainable AI extends beyond Logistic Regression, as it helps ensure fairness, compliance, and ethical decision-making in finance (Arrieta et al., 2020; Brundage et al., 2018). Interpretability and explainability may be legally required in some regulatory environments to protect borrowers and promote responsible lending practices (Goodman and Flaxman, 2017; Selbst and Powles, 2018; Chen et al., 2018; Bussmann et al., 2020; Demajo et al., 2020; Misheva et al., 2021).

In light of the importance of explainability and interpretability, various methods have been developed to improve the accuracy of a single classifier, such as Logistic Regression. These include feature scaling, feature engineering, regularization, hyperparameter tuning, and the cluster-then-predict approach (Hastie et al., 2009; James et al., 2013; García et al., 2015; Géron, 2022). The cluster-then-predict approach groups similar instances to build models for

each group, partitioning data into homogeneous clusters based on similarity, and then training a separate model for each cluster. Clustering-then-predict has been applied in various domains, including finance, to improve overall accuracy (Peikari et al., 2018; Radu et al., 2014; Tsai, 2014; Chen and Shyu, 2011).

The scaling of features significantly impacts clustering, as clustering algorithms such as Euclidean distance depend on the values of the features in the dataset. Unlike standard scaling techniques such as min-max normalization and z-score scaling that treat every feature equally, the proposed rescaled cluster-then-predict approach rescales features based on their impact on the target. This rescaling allows for a similarity or distance measure that better reflects the essential significance of each feature. The objective of rescaling is to enhance the data representation by emphasizing variation for important features and reducing variation for less significant features.

Building credit scoring models may encounter imbalanced data, where default (positive) cases are fewer than non-default (negative) cases. Solutions to this problem include adjusting the default threshold, using relevant evaluation metrics, applying machine learning algorithms suited for imbalanced data, cost-sensitive classification, and sampling methods (He and Garcia, 2009; Sun et al., 2009; Krawczyk, 2016; Fernández et al., 2018; Aggarwal, 2015). The importance of accurately identifying positive cases thus motivates us to investigate the impact of including only default cases versus all cases when clustering.

Focusing on the efficacy of clustering strategies for imbalanced data, our research propels the evolution of credit scoring models that are both precise and interpretable. We introduce four rescaling methods that are applied to preprocessed data, followed by clustering into subgroups, where distinct models are built for each. To scrutinize the potency of this "rescaled cluster-then-predict" methodology in enhancing predictions, we employ two machine learning algorithms - the cutting-edge XGBoost and the intuitive Logistic Regression. Through the analysis of two diverse datasets, we observe that XGBoost's predictions do not benefit from our approach, and even experience a diminished Area Under the Curve (AUC) as the cluster count rises and polynomial transformed features are incorporated. However, Logistic Regression exhibits an escalated AUC when a particular rescaling method is implemented for clustering, and polynomial features are integrated. Notably, clustering solely the positive cases is as effective as clustering the entire dataset, signifying substantial gains in computational efficiency.

The rest of this paper is organized as follows. Section 2 addresses motivations and describes the rescaled cluster-then-predict approach. Data analysis on two datasets is separately presented in Sections 3 and 4, demonstrating the effectiveness of our approach in handling imbalanced data and improving the interpretability of credit scoring models. Finally, the last section concludes and provides directions for future research in the field of credit scoring and explainable AI.

2. Motivations and features resacling

In this section, we delve into the process of data preprocessing, setting the stage for our rescaled, cluster-then-predict approach. To substantiate the potency of our methodology, we bring forward two unique datasets, acting as our motivations and practical demonstrations.

2.1. Data preprocessing

To ensure a balanced comparison between the prediction performances of XGBoost and Logistic Regression, we employ a series of data preprocessing steps:

1. Log Transformation: Given that Logistic Regression is sensitive to extreme values, we apply log transformation to features exhibiting high kurtosis (greater than 10) to mitigate the influence of outliers.
2. Z-score Normalization: To circumvent potential round-off errors and numerical instabilities during the estimation of Logistic Regression parameters, we normalize non-binary features using Z-score scaling. This ensures that the features have a mean of zero and a standard deviation of one.
3. Handling Missing Values with Dummy Variables: Instead of discarding observations with missing values, which may result in information loss, we introduce dummy variables to indicate the presence of missing data. In the context of Logistic Regression, the inclusion of a dummy variable effectively bifurcates the dataset into two subsets – one comprising observations with complete data and the other with missing values. Notably, both subsets share the same model parameters with the exception of the intercept, allowing for more comprehensive utilization of the available data.

These preprocessing steps are crucial in optimizing the data for the subsequent application and evaluation of XGBoost and Logistic Regression models.

2.2. Features rescaling

The proposed rescaled cluster-then-predict approach rescales features based on their impact on the target to produce a similarity or distance measure that better reflects the significance of each feature. This enhances data representation by emphasizing and increasing variation for important features and reducing variation for less significant ones.

Suppose we have n observations in our dataset, each with p features as explanatory variables. For the i-th observation, its feature vector is denoted as $x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_p^{(i)})$ and its target value is denoted as $y^{(i)}$ for $i = 1, \ldots, n$. The goal of rescaling is to adjust the influence of the features on the target by finding the rescalor $w = (w_1, \ldots, w_p)$. Once w is determined, we rescale each feature by multiplying it by the corresponding weight:

$$\tilde{x}_j^{(i)} = w_j x_j^{(i)},$$

for $j = 1, \ldots, p$ and $i = 1, \ldots, n$. We propose four methods for determining the rescalor w.

2.2.1. The Equal Weight rescalor

The Equal Weight (EW) rescalor is simply a vector of ones, $w = (1, 1, \cdots, 1)$. The EW rescalor is simply serves as a benchmark, so that we can compare if other rescalors lead to a clustering scenario that helps to improve the prediction.

2.2.2. The Linear Regression rescalor

For the Linear Regression (REG) rescalor, we use the coefficients from multiple linear regression to determine the rescalor. The multiple linear regression model is

$$y^{(i)} = \beta_0 + \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \cdots + \beta_p x_p^{(i)} + \varepsilon_i,$$

where the error ε_i is normally distributed with mean zero and variance σ^2. The coefficients $\beta_0, \beta_1, \ldots, \beta_p$ are estimated by the least-squared estimation. The rescalor w_j equals the estimated value of β_j for $j = 1, \ldots, p$.

2.2.3. The Logistic Regression rescalor

For the Logistic Regression (LR) rescalor, we consider the logistic regression because the target variable is binary in the credit scoring model. Let

Bernoulli(π) denote the Bernoulli distribution with probability π. In the Logistic Regression model, we assume

$$y^{(i)}|x^{(i)} \sim Bernoulli\left(\frac{1}{1+e^{-\left(\beta_0+\beta_1 x_1^{(i)}+\beta_2 x_2^{(i)}\cdots+\beta_p x_p^{(i)}\right)}}\right),$$

where the maximum likelihood estimation estimates the coefficients. The rescalor w_j equals the estimated value of β_j for $j = 1, \ldots, p$.

2.2.4. The Mutual Information rescalor

The Mutual Information (MI) rescalor is motivated by the mutual information rescalor, which measures the dependency between the target variable y and the j-th feature x_j. Let $P_{x_j}(a)$ be the sample marginal probability of x_j taking on the value a, and $P_y(b)$ and $P_{x_j,y}(a, b)$ be similarly defined.

If x_j is a discrete feature, the mutual information is calculated as follows:

$$MI(x_j, y) = \sum_{(a,b)} P_{x_j,y}(a, b) \log \frac{P_{x_j,y}(a, b)}{P_{x_j}(a)P_y(b)}, \tag{1}$$

where (a, b) runs over pairs of $(x_j^{(i)}, y^{(i)})$ for all i.

When x_j is a continuous feature with numerical values, we estimate the mutual information using the k-nearest neighbor method with $k = 3$ proposed by Ross (2014). Given an i, define d as the distance between $x_j^{(i)}$ and its k-nearest neighbor with the same target value as $y^{(i)}$. Let $n_{y^{(i)}}$ denote the sample proportion of the target set having the same value as $y^{(i)}$, and m_i denote the number of samples in $x_j^{(1)}, \ldots, x_j^{(n)}$ with distance to $x_j^{(i)}$ less than or equal to d. The formula for mutual information is:

$$MI(x_j, y) = \frac{1}{n}\sum_{i=1}^{n} \left[\psi(n) - \psi(n_{y_i}) + \psi(k) - \psi(m_i)\right], \tag{2}$$

where $\psi(t) = \frac{\Gamma'(t)}{\Gamma(t)} \approx \ln t - \frac{1}{2t}$ and $\Gamma(t) = (t-1)!$ for a non-negative interger t. The mutual information rescalor for each feature is its mutual information with the target variable using (1) or (2), depending on whether it is discrete or continuous.

2.3. Example 1: The PAK dataset

The PAK dataset is a private dataset from a Brazilian credit company and its partner shops, which is avaiable at `https://pakdd.org/archive/pakdd2010/PAKDDCompetition.html` It contains 50,000 cases, with a target variable indicating default and 53 features, of which 40 are discrete, and 13 are continuous. The dataset also has an imbalanced data problem with a default rate of 26.1%. For detailed information regarding the PAK dataset, please refer to Appendix A. The final variables considered for subsequent analysis consist of the binary target variable y and the original features x_1 to x_{12}.

With these features, we employ Principal Component Analysis (PCA) as a dimensionality reduction technique to efficiently visualize the multidimensional data. In Figure 1, the data is projected onto a two-dimensional plane defined by the first and second principal components. This visualization reveals a natural segregation of the data into four distinct clusters. Motivated by this observation, we hypothesize that developing separate predictive models for each of these clusters could enhance the overall predictive accuracy. The rationale behind this hypothesis is that data points within each cluster may share certain underlying characteristics, which could be exploited by a specialized model to make more accurate predictions.

Nonetheless, our preliminary analysis reveals that predictions based on clusters formed using these features do not result in a significant improvement in prediction accuracy. It is posited that this lack of improvement might be attributed to the application of Z-score normalization, which could be concealing the intrinsic importance of individual features by standardizing their scale. To address this issue, we propose alternative rescaling methods for the features in the subsequent sections, aimed at preserving their inherent significance and enhancing the predictive accuracy of the models.

Utilizing the PAK dataset, we embark on visualizing the features rescaled through the REG, LG, and MI methods using PCA. The visualizations are depicted in Figure 1. Interestingly, a comparative analysis of these visualizations with those of the non-rescaled features (EW-rescaled) reveals markedly distinct patterns. Specifically, REG- and LR-rescaled features manifest six discernible clusters, whereas the MI-rescaled features exhibit a bifurcated structure with only two clusters. These differences underscore the impact of rescaling methods on the data distribution and clustering patterns.

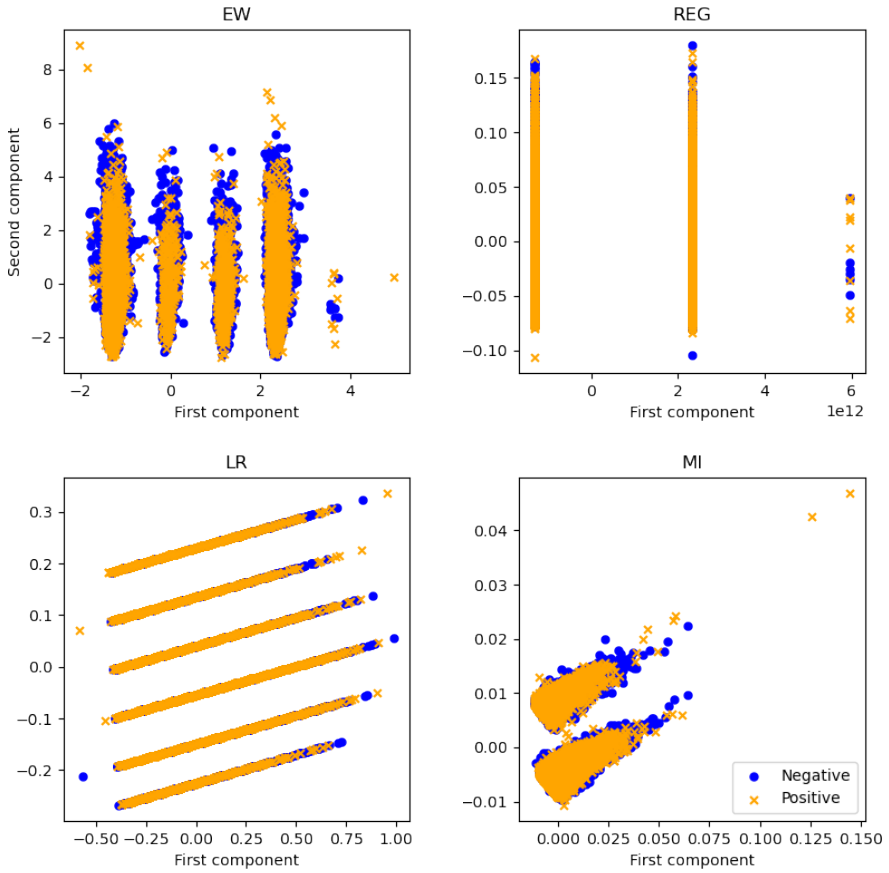


Figure 1: PCA visualizations of the PAK dataset with various rescaling methods. Each subfigure represents the data projected onto the first and second principal components after applying a specific rescaling method.

2.4. Example 2: The GMC dataset

In our second experiment, we employ the 'Give Me Some Credit' (GMC) dataset, sourced from Kaggle, which comprises ten features alongside a target variable denoting credit defaults. The dataset is accessible at `https://www.kaggle.com/competitions/GiveMeSomeCredit/data`. It is noteworthy that the dataset is imbalanced, containing 150,000 observations, of which only 10,026 instances are credit defaults, translating to a default rate of approximately 6.7%. Detailed information regarding data preprocessing can be found in Appendix B. The processed dataset encompasses a target variable, y, and 12 features, denoted as $x_1, \ldots, x_{12}$.

In contrast to the distinct clustering observed in the PAK dataset (as depicted in Figure 2), the GMC dataset exhibits a different structure, as illustrated in Figure 1. Specifically, we observe four clusters, which are not clearly separated but rather exhibit overlaps. This serves as an alternative scenario and provides an opportunity to evaluate the robustness and versatility of our proposed cluster-then-predict approach in handling datasets with different characteristics. In the ensuing sections, we undertake a comprehensive analysis of these clusters and assess the impact of the observed patterns on predictive modeling."

Figure 2 also illustrates the PCA visualizations of the REG-, LR-, and MI-rescaled features. A striking divergence in clustering patterns is observed when compared to the visualizations of the unscaled (EW-rescaled) features. In particular, the REG-scaled features give rise to approximately eight clusters, the LR-scaled features manifest close to 16 clusters, while the MI-scaled features yield around 18 clusters. These variations in clustering underscore the profound impact of feature scaling on the underlying data structure, which in turn holds implications for subsequent predictive modeling.

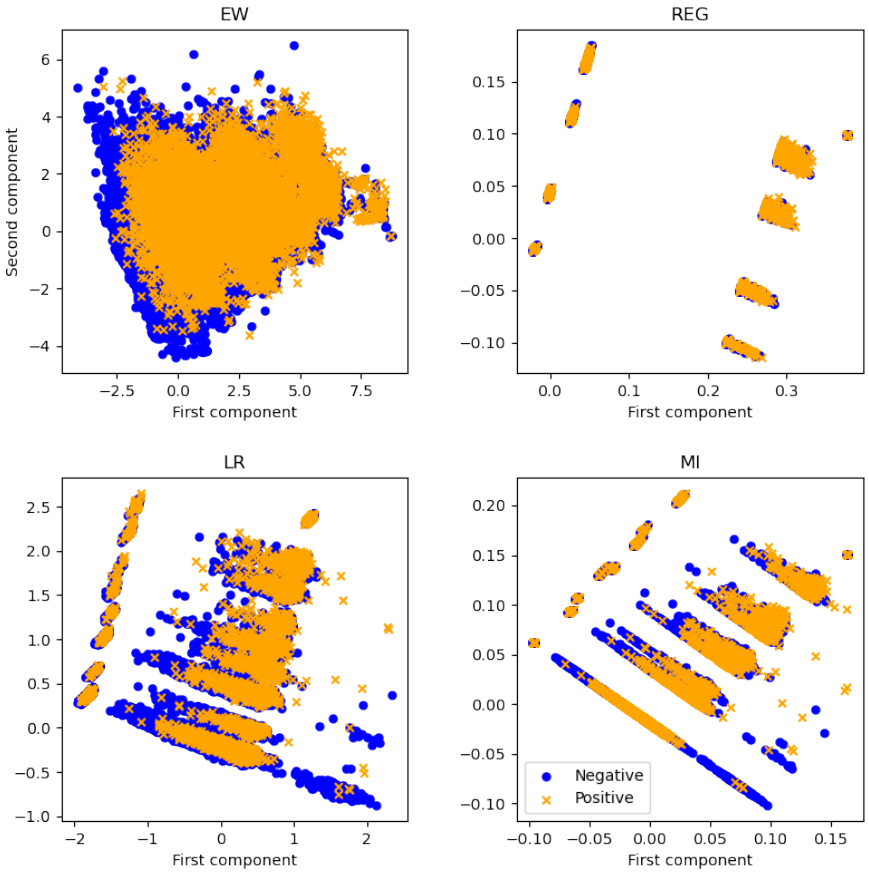


Figure 2: PCA visualizations of the GMC dataset with various rescaling methods. Each subfigure represents the data projected onto the first and second principal components after applying a specific rescaling method.

3. Experiment of the PAK data

In the feature engineering phase, quadratic and interaction terms are integrated into the model. Specifically, we incorporate x_i, x_i^2, and $x_i x_j$ terms for $i = 1, \ldots, p$, where $0 \leq i < j \leq p$ and p represents the number of features ($p = 12$ in the PAK dataset). To circumvent collinearity, cubic features are excluded from the analysis. The dataset is subsequently segmented into an 80-20 split for training and testing, respectively.

We proceed by computing four rescaling methods - EW, REG, LR, MI - and applying them to the features. The K-medoids clustering algorithm is adopted owing to its superiority in robustness compared to K-means (Choi and Kwon, 2015). Our preliminary analysis encompasses experiments with clusters ranging from $K = 2$ to $K = 6$, where $K = 1$ signifies no clustering. This stage is vital for assessing how clustering impacts default rates and entropy reduction within clusters.

As algorithms for classification, XGBoost is acknowledged for its advanced performance in credit scoring, while Logistic Regression is employed for its traditional simplicity in binary classification. We examine performance with $K = 1, 2, 3, 4$ clusters, where $K = 1$ denotes no clustering. The AUC metric, reflecting the area under the Receiver Operating Characteristic Curve, is used for evaluating classifier performance.

We outline our methodology as follows:

1. Data Preprocessing and Feature Engineering:
   (a) Preprocess features through logarithmic transformation, z-score normalization, and missing value imputation. The outcome is termed 'original features.'
   (b) Generate quadratic transformations of the original features, referred to as 'quadratic features.'
   (c) Compute EW, REG, LR, and MI rescalers for both original and quadratic features to derive rescaled features.
   (d) Partition the dataset into 80% training and 20% testing subsets.
2. Execute K-medoids Clustering:
   (a) Observe default rates for each cluster.
   (b) Investigate both the entropy of individual clusters and total entropy.
3. Develop K sub-models for Each Cluster using XGBoost and Logistic Regression:

(a) Allocate new samples to the nearest cluster and use the relevant sub-model for prediction.
(b) Compute AUC scores for each configuration on both the training and testing datasets.

3.1. Data exploration and clustering

Figure 3 depicts the default rates for varying numbers of clusters ($K = 2, \ldots, 6$). Solid circles represent the training set, while hollow circles represent the testing set. The overall default rate of 26.1% is denoted by a horizontal black line. Notably, the default rates within individual clusters diverge from the overall default rate, as evidenced in both the training and testing datasets. Moreover, a close alignment is observed between default rates of individual clusters in the training and testing sets.

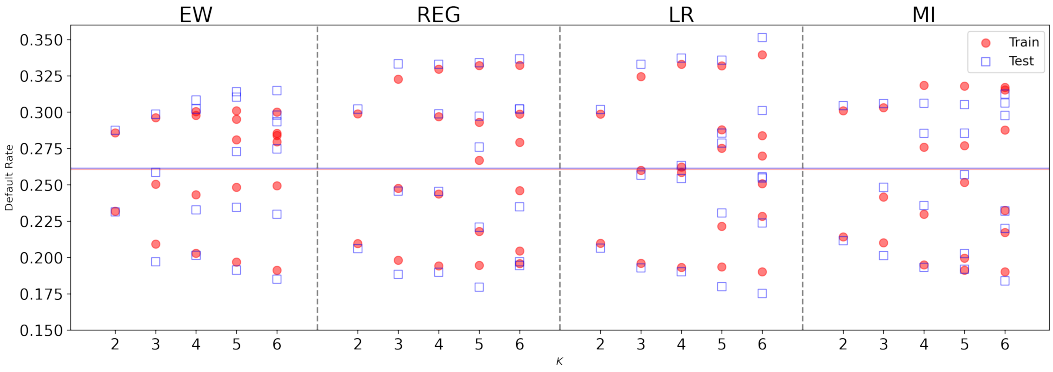


Figure 3: Default rates comparison for $K = 2, \ldots, 6$ clusters using EW-, REG-, LG-, and MI-rescaled features in the PAK dataset. Solid circles and hollow squares denote training and testing sets, respectively. The horizontal black line represents the overall default rate of 26.1%.

We further evaluate the efficacy of our clustering methodology by computing the total entropy, a metric rooted in information theory that quantifies the uncertainty or information content of a random variable or message Gray (2011). For the i-th cluster with a default rate represented as p_i, the entropy is calculated as:

$$h_i = -\left(p_i \log(p_i) + (1 - p_i) \log(1 - p_i)\right).$$

A lower value of entropy within a cluster signifies a more concentrated distribution of default cases, indicative of higher proportions of either positive

or negative instances. The total entropy is computed as the weighted sum of the individual entropies across K clusters, with the weights being the respective proportions of the clusters in the dataset:

$$h = \sum_{i=1}^{K} q_i h_i,$$

where q_i denotes the fraction of data points in the i-th cluster. A lower total entropy points to more concentrated default cases within clusters, reflecting the efficacy of the clustering approach.

Figure 4 illustrates the entropy values for different configurations in the PAK dataset with the number of clusters ranging from $K = 2$ to $K = 6$. The figure displays the entropy values for the EW-, REG-, LR-, and MI-reseacled features with solid lines representing the training dataset and dashed lines representing the testing dataset. The horizontal black solid and dashed lines indicate the entropy values for the training and testing datasets, respectively, in the scenario where $K = 1$, i.e., without clustering. The similarity between the entropy values for the training and testing sets substantiates the reliability of the entropy metric and the consistency across clusters. Notably, as the number of clusters increases, the entropy tends to decrease for both clustering methods when utilizing original features. We exclude the results for REG-rescaled quadratic features from further analysis as clustering proved ineffective.

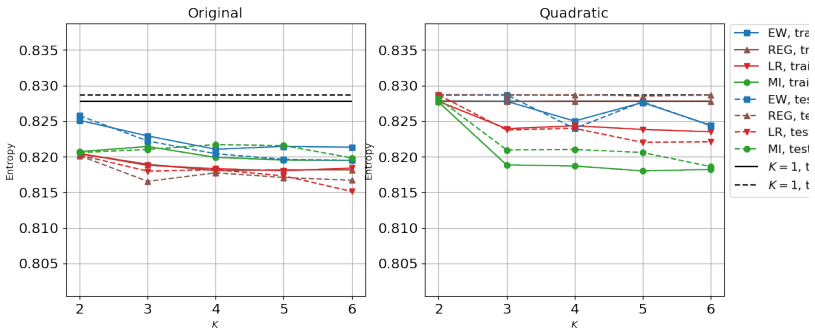


Figure 4: Comparison of total entropies for $K = 2, \ldots$ clusters, utilizing EW-, REG-, LG-, and MI-rescaled features in the PAK dataset. Solid and dashed lines represent training and testing sets, respectively.

3.2. Evaluating the impact of clustering on prediction accuracy

Figure 5 contrasts the Area Under the Curve (AUC) obtained through XGBoost and Logistic Regression on training and testing sets when applying the EW, REG, LR, and MI rescalors with varying numbers of clusters ($K = 1, 2, 3, 4$) based on the original feature set.

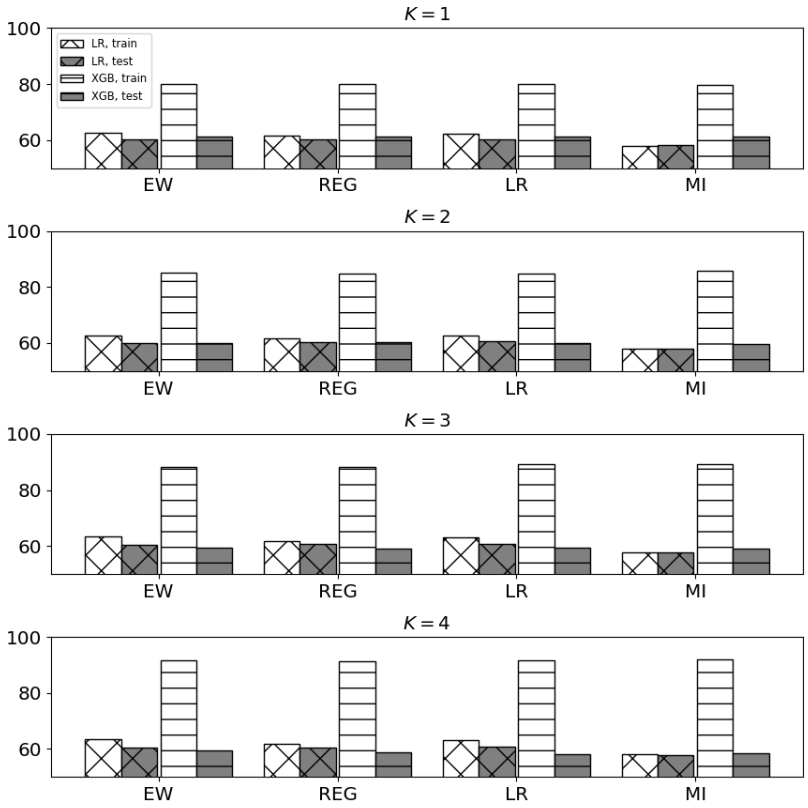


Figure 5: Comparative analysis of AUC between XGBoost and Logistic Regression across various cluster sizes ($K = 1, 2, 3, 4$) using the EW, REG, LR, and MI rescaling on the PAK dataset

Initially, for the original feature set and $K = 1$ (indicating no clustering), XGBoost exhibits a substantially higher AUC in the training set, whereas a diminished AUC is observed in the testing set, hinting at overfitting. Conversely, Logistic Regression yields consistent AUC values across both sets, indicating stability. Notably, as K increases, the discrepancy in AUC between the training and testing sets for XGBoost widens, further accentuating its

overfitting tendency. Logistic Regression, however, maintains its stability with increasing K. Shifting focus to the AUC in the testing set, XGBoost does not demonstrate an increase in AUC with escalating K across any of the rescaled features (EW-, REG-, LR-, MI-).

This is further evidenced when employing quadratic features; XGBoost exhibits a significantly elevated AUC in the training dataset compared to the testing dataset, reaffirming the overfitting issue. Contrastingly, Logistic Regression consistently exhibits comparable AUC values for training and testing datasets across all rescaled features with quadratic transformations, indicating robustness. Figures depicting these consistent findings for Logistic Regression have been omitted for the sake of brevity.

This analysis reveals that clustering does not inherently improve the predictive accuracy of XGBoost, which exhibits a susceptibility to overfitting, particularly as the complexity of features and number of clusters increases. On the other hand, Logistic Regression demonstrates stable performance and robustness across various feature rescaling techniques and cluster sizes.

We rigorously compare the AUC on the testing dataset to discern if clustering enhances performance. Table 1 presents the AUC for the testing set employing both the original and quadratic feature sets. As a values, for hte original feature set with the EW rescaling (indiciating no scaling), the XGBoost yield an AUC of value 60.59, and the Logistic Regression yields an AUC of value 60.16. Although the XGBoost produces higher AUC, it is simply a 0.71% change compared with the Logistic Regression.

For the original feature set, XGBoost attains a superior AUC compared to Logistic Regression when $K = 1$. The peak AUC for XGBoost, 61.32, is achieved with the MI-rescaled features. XGBoost's AUC exhibits a decline as K ascends. With the quadratic feature set, XGBoost's AUC is diminished compared to the original features and continues to wane as the dataset is partitioned into more clusters (i.e., with increasing K). For the original feature set, XGBoost attains a superior AUC compared to Logistic Regression when $K = 1$. The peak AUC for XGBoost, 61.32, is achieved with the MI-rescaled features. XGBoost's AUC exhibits a decline as K ascends. With quadratic features, XGBoost's AUC is diminished compared to the original features and continues to wane as the dataset is partitioned into more clusters (i.e., with increasing K).

Conversely, Logistic Regression's AUC escalates with increasing K for the original feature set. When employing quadratic features, Logistic Regression surpasses XGBoost at $K = 1$ for the EW-, REG-, and LR-rescaled features,

Table 1: AUC measurements using EW, REG, LR, and MI rescalors applied to the original and quadratic feature sets in the PAK dataset's testing set. The benchmark AUCs for XGBoost and Logistic Regression, derived when $K = 1$ and utilizing the EW rescalor, are underlined. The highest AUC values for both XGBoost and Logistic Regression are highlighted in bold. For Logistic Regression, an asterisk denotes configurations that yield an AUC exceeding the maximum AUC attained by XGBoost.

	XGBoost				Logistic Regression			
K	EW	REG	LR	MI	EW	REG	LR	MI
The original feature set								
1	<u>60.59</u>	61.20	61.31	**61.32**	<u>60.16</u>	58.22	60.14	60.19
2	60.59	59.57	60.07	60.15	60.16	57.90	60.56	60.29
3	60.59	59.16	59.40	59.19	60.16	57.81	60.86	60.75
4	60.59	58.63	57.98	58.76	60.16	57.84	60.96	60.32
The quadratic feature set								
1	59.94	59.94	59.94	60.38	61.45*	**62.00***	61.99*	59.96
2	59.07		59.07	59.25	60.60		61.70*	59.69
3	58.81		58.57	59.03	60.01		61.93*	59.49
4	58.50		58.30	58.43	58.96		61.81*	59.09

signifying that quadratic transformations are beneficial for Logistic Regression. Generally, Logistic Regression maintains or increases its AUC with the escalation of K, especially with the LR rescaling, and proves superior to XGBoost for $K = 2, 3$. Notably, with the quadratic feature set and LR rescaling, Logistic Regression reaches an AUC of 61.99 at $K = 1$, a 1.11% enhancement. As K increases, Logistic Regression with LR rescaling sustains an AUC exceeding XGBoost's peak AUC. These observations reveal that quadratic transformations bolster Logistic Regression's performance, while clustering has a positive impact on Logistic Regression and a detrimental effect on XGBoost. Moreover, with appropriate rescaling, Logistic Regression can outshine XGBoost in terms of AUC.

In summary, Table 1 establishes that the efficacy of clustering in enhancing predictions is contingent upon (1) the machine learning algorithm, and (2) the rescaling method employed prior to clustering. XGBoost's AUC diminishes with additional clusters and quadratic features, whereas Logistic Regression's AUC thrives, particularly when the LR rescaling is applied.

3.3. Evaluating clustering strategies: All cases versus positive cases

Given that credit scoring models frequently grapple with class imbalance, this section evaluates and compares two distinct clustering approaches: all-case clustering and positive-case clustering. Both employ the K-medoids algorithm, albeit with differing strategies in determining medoids. Specifically, all-case clustering utilizes all cases for medoid identification, whereas positive-case clustering solely relies on positive cases. The rationale for the latter approach is grounded in the heightened importance of accurately identifying the positive cases, which often constitute the minority class. We conduct a comprehensive analysis to compare the outcomes of these clustering strategies.

Interestingly, both the default rates and entropies observed with positive-case clustering are akin to those with all-case clustering. This similarity in results also holds when employing quadratic features; therefore, we have elected to omit the corresponding figure for the sake of conciseness.

Table 2 juxtaposes the AUC obtained from XGBoost and Logistic Regression through positive-case clustering. This comparison incorporates EW-, REG-, LR-, and MI-rescaled original and quadratic feature sets for the test dataset. When contrasted with Table 1, both positive-case and all-case clustering strategies exhibit analogous AUCs. This suggests that focusing on clustering solely the positive cases, as opposed to all cases, attains comparable predictive accuracy.

In terms of XGBoost, incorporating quadratic features or escalating the number of clusters, even when features are rescaled, does not augment the AUC. Conversely, Logistic Regression demonstrates an enhancement in AUC when employing the quadratic feature set. Notably, with LR-rescaled features applied to both original and quadratic feature sets, Logistic Regression progressively increases its AUC as the value of K escalates. However, when using EW-, REG-, and MI-rescaled features, Logistic Regression's AUC remains relatively invariant. This implies that the selection of an apt rescaling technique is pivotal in harnessing the benefits for Logistic Regression.

Table 2: AUC values utilizing the EW, REG, LR, and MI rescalers with positive-case clustering applied to the original and quadratic feature sets in the PAK dataset's testing set. The peak AUC for XGBoost and Logistic Regression are highlighted in bold. An asterisk beside values for Logistic Regression indicates configurations that outperform the highest AUC achieved by XGBoost.

	XGBoost				Logistic Regression			
K	EW	REG	LR	MI	EW	REG	LR	MI
The original feature set								
1	<u>60.59</u>	61.20	61.31	**61.32**	<u>60.16</u>	58.22	60.14	60.19
2	60.59	59.33	59.48	60.00	60.16	58.26	60.56	60.22
3	60.59	59.90	59.24	59.78	60.16	57.32	60.50	60.16
4	60.59	59.18	58.81	59.01	60.16	57.79	61.07	60.44
The quadratic feature set								
1	59.94	59.94	59.94	60.38	61.45*	62.00*	**61.99***	59.96
2	58.84		59.07	59.33	60.08		61.78*	59.77
3	59.11		58.04	58.77	59.58		61.89*	59.34
4	57.95		58.00	57.55	59.42		61.77*	58.79

4. Experiment of the GMC dataset

For the GMC dataset, we adopt the methodology outlined in Section 3 and extend it by incorporating cubic features and interactions. Specifically, the cubic feature set enhances the quadratic set by appending x_i^3, $x_i^2 x_j$, $x_i x_j^2$, and $x_i x_j x_k$ for $i = 1, \ldots, p$, $1 \leq i < j \leq p$, and $1 \leq i < j < k \leq p$. We employ the rescaled cluster-then-predict approach and juxtapose the outcomes of positive-clustering with all-case clustering.

4.1. Data exploration and clustering

Figure 6 illustrates the default rates across clusters for $K = 2$ to $K = 6$ using solid circles for the training set and hollow squares for the testing set. A horizontal black line denotes the aggregate default rate of 6.7%. The proximity of solid and hollow symbols indicates that training and testing sets yield consistent and homogeneous clusters with comparable default rates. Similar observations hold for quadratic and cubic feature sets, so additional figures are omitted for brevity.

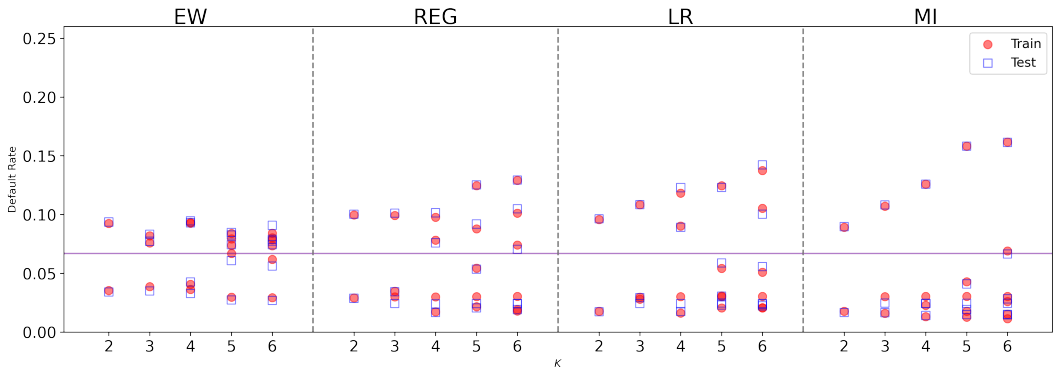


Figure 6: Default rates comparison for $K = 2, \ldots, 6$ clusters using EW-, REG-, LG-, and MI-rescaled features in the GMC dataset. Solid circles and hollow squares denote training and testing sets, respectively. The horizontal black line represents the overall default rate of 6.7%.

Figure 7 portrays the entropy of clustering all cases for $K = 2, \ldots, 6$ after applying EW, REG, LG, and MI rescaling to the original feature set. The solid and dashed lines correspond to the entropy computed from the training and testing sets, respectively. Their similarity suggests a consistency in the clustering behavior across the training and testing sets.

As K increases, entropy diminishes, implying a reduction in data randomness with a higher number of clusters. Upon closer examination, we observe that clustering the EW-rescaled features results in higher entropy compared to REC-, LG-, and MI- rescaled features, suggesting that EW-rescaled features are less effective in minimizing randomness.

Both Figures 6 and 7 illustrate that rescaled feature clustering can potentially decrease data randomness. This prompts a further investigation into whether constructing separate models based on these clusters can enhance prediction accuracy.

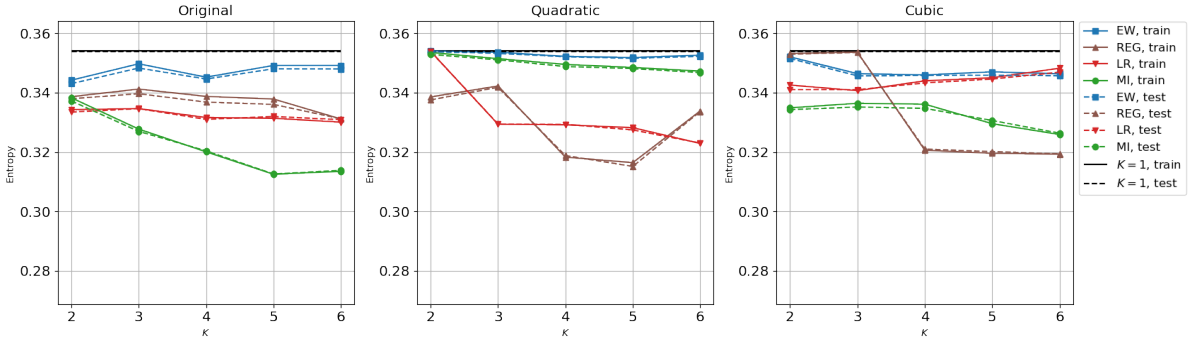


Figure 7: Comparison of total entropies for $K = 2, \ldots$ clusters, utilizing EW-, REG-, LG-, and MI-rescaled features in the GMC dataset. Solid and dashed lines represent training and testing sets, respectively.

4.2. Evaluating the impact of clustering on prediction accuracy

Figure 8 allows us to examine whether XGBoost or Logistic Regression is prone to overfitting. Each row presents the AUC for varying cluster numbers, $K = 1, 2, 3, 4$. XGBoost exhibits a significantly higher AUC in the training data than the testing data, suggesting overfitting, while Logistic Regression demonstrates robustness. This pattern remains consistent for $K = 1, 2, 3, 4$.

Even with the quadratic and cubic feature sets, XGBoost continues to show signs of overfitting, evidenced by a notably higher AUC in the training set compared to the testing set. Meanwhile, Logistic Regression maintains its robustness, delivering similar AUCs in both training and testing sets. These observations further underscore XGBoost's overfitting issue and Logistic Regression's reliability. As the AUCs for the quadratic and cubic feature sets closely resemble Figure 8, they have been omitted for brevity.

Table 3 presents a comparison of the AUC between XGBoost and Logistic Regression on the testing set, using original, quadratic, and cubic feature sets, each rescaled by EW, REG, LR, and MI methods.

For the original feature set, while XGBoost yields a higher AUC than Logistic Regression for $K = 1$, further clustering does not enhance its AUC. As K increases, XGBoost's AUC declines. This pattern holds true for the quadratic and cubic feature sets, underscoring the difficulty in augmenting XGBoost performance using polynomial transformations or clustering, characteristic of its nature as an ensemble method.

Contrarily, Logistic Regression, with the original feature set, demonstrates an increase in AUC as K increases when EW and LR rescaling methods are used. In the context of the quadratic feature set, despite XGBoost's superior AUC at $K = 1$, an increase in K leads to a decline in XGBoost's AUC and an elevation in Logistic Regression's AUC, especially with LR rescaled features, which achieve the highest AUC at $K = 2, 3, 4$.

In the context of the cubic feature set, XGBoost initially outshines Logistic Regression when using EW, REG, and MI rescaled features at $K = 1$. However, as we increment K, the AUC for Logistic Regression enhances, reaching its zenith with LR rescaled features at $K = 2, 3, 4$. Notably, the maximum AUC is yielded by Logistic Regression when the LR rescalor is utilized with $K = 3$, resulting in an AUC of 85.52. This figure stands in close proximity to the highest AUC produced by XGBoost, which is 85.73.

Table 3 also substantiates that the efficacy of clustering for prediction enhancement hinges on (1) the selected machine learning algorithm and (2)

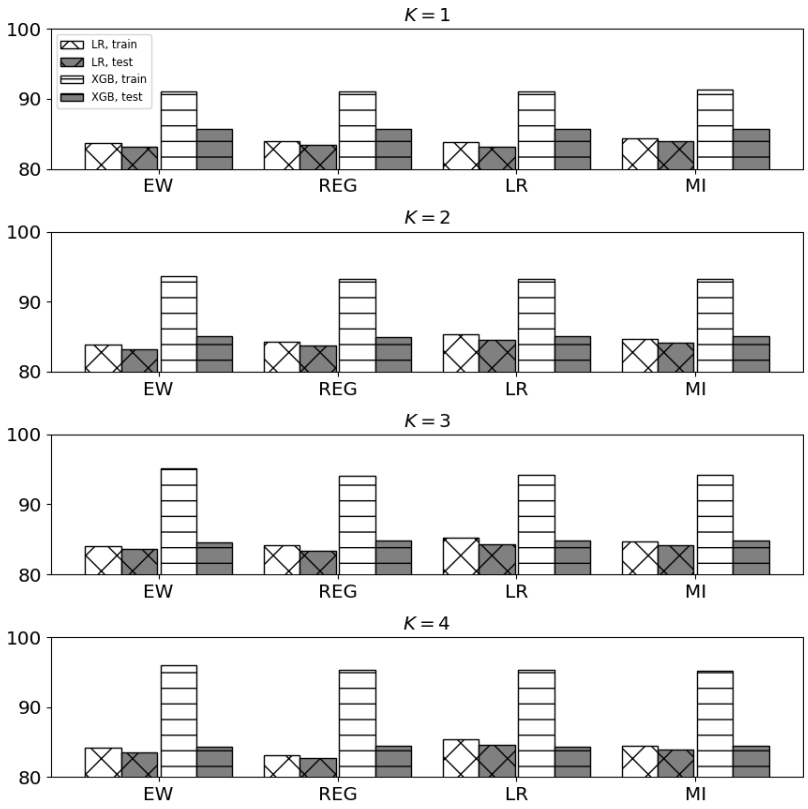


Figure 8: Comparative analysis of AUC between XGBoost and Logistic Regression across various cluster sizes ($K = 1, 2, 3, 4$) using the EW, REG, LR, and MI rescaling on the PAK Dataset

the rescaling method employed before clustering. It highlights the decrease in XGBoost's AUC with an increase in clusters or complexity of quadratic and cubic features, while Logistic Regression's AUC improves. Of note, Logistic Regression with LR-rescaled features consistently outperforms its counterparts rescaled with EW, REG, and MI.

Table 3: AUC measurements using EW, REG, LR, and MI rescalors applied to the original, quadratic, and cubic feature sets in the GMC dataset's testing set. The benchmark AUCs for XGBoost and Logistic Regression, derived when $K = 1$ and utilizing the EW rescalor, are underlined. The highest AUC values for both XGBoost and Logistic Regression are highlighted in bold. For Logistic Regression, a plus denotes a configuration that yield the highest AUC.

	XGBoost				Logistic Regression			
K	EW	REG	LR	MI	EW	REG	LR	MI
The original feature set								
1	<u>85.66</u>	**85.73**	85.66	85.66	<u>83.17</u>	**84.02**	83.19	83.48
2	85.07	85.28	85.01	84.87	83.12	84.14	84.54	83.68
3	84.56	84.89	84.87	84.82	83.61	84.14	84.27	83.42
4	84.32	84.48	84.35	84.51	83.48	83.92	84.56	82.66
The quadratic feature set								
1	85.35	85.24	85.24	85.23	84.36	84.43	84.51	84.46
2	84.36	84.68	84.43	84.53	84.50	84.30	85.11	84.50
3	83.51	84.09	84.43	83.83	84.54	84.48	85.23	84.37
4	83.51	83.56	84.02	83.82	84.39	84.54	85.40	84.34
The cubic feature set								
1	84.87	84.87	84.77	84.87	83.44	84.72	85.26	82.51
2	84.05	84.17	83.78	84.24	83.67	84.78	85.51	82.55
3	83.59	83.24	83.15	83.49	83.44	84.75	85.52^{+}	82.63
4	82.65	82.85	82.60	82.86	83.48	84.81	85.42	84.04

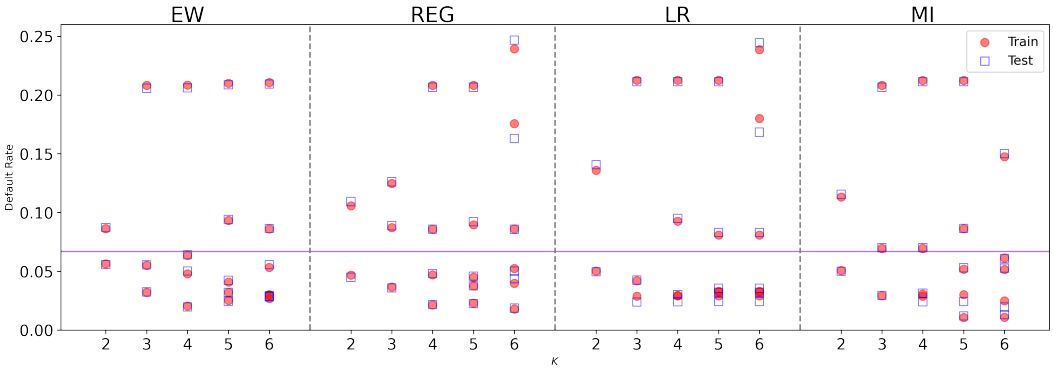


Figure 9: Default rates comparison for $K = 2, \ldots, 6$ clusters using EW-, REG-, LG-, and MI-rescaled features in the GMC dataset when clustering only positive cases. Solid circles and hollow squares denote training and testing sets, respectively. The horizontal black line represents the overall default rate of 6.7%.

4.3. Evaluating clustering strategies: All cases versus positive cases

As seen in Section 3.3, we will now assess whether clustering solely positive cases can achieve a predictive accuracy akin to that obtained by clustering all cases. The default rates when implementing EW, REG, LR, and MI-rescaled original feature set clustering in the GMC dataset for $K = 2, \ldots, 6$ are depicted in Figure 9. It is important to note that clustering based on positive cases results in a broader variety of default rates as opposed to all-case clustering, as illustrated in Figure 6.

The entropy reduction using positive clustering proves to be more substantial compared to all-case clustering as seen in Figure 4: The entropy drops to 0.28 when only positive cases are clustered, whereas it descends to about 0.32 in the case of all-case clustering. As evidenced by Figures 9 and 10, exclusively clustering positive cases with rescaled features enhances the reduction in randomness to a notable extent.

Table 4 provides a comparison of AUC between XGBoost and Logistic Regression on the testing set, utilizing the original, quadratic, and cubic feature sets, each rescaled by EW, REG, LR, and MI methods when clustering only positive cases. This corroborates our conclusion from Table 3: XGBoost does not yield higher AUC values when using the quadratic or cubic feature sets, nor when K is increased. In contrast, Logistic Regression exhibits higher AUC values with the quadratic and cubic feature sets, particularly when the LR rescaling method is applied with increasing K. The highest achieved

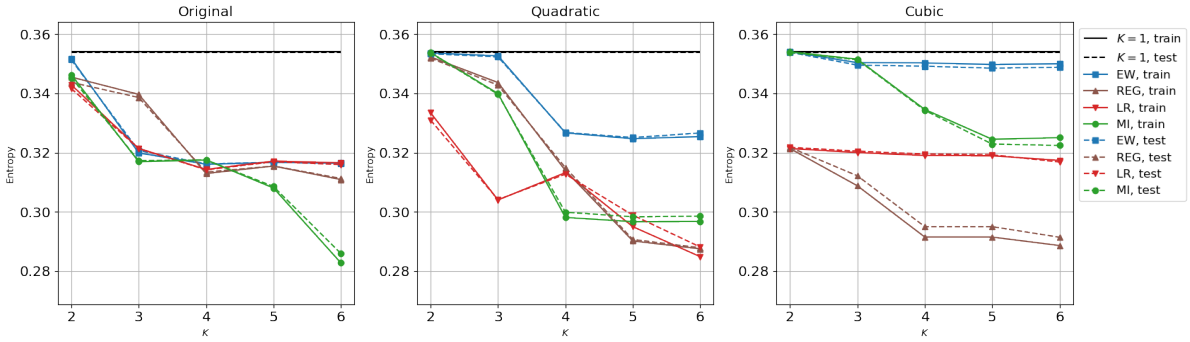


Figure 10: Comparison of total entropies for $K = 2, \ldots$ clusters, utilizing EW-, REG-, LG-, and MI-rescaled features in the GMC dataset when clustering only positive cases. Solid and dashed lines represent training and testing sets, respectively.

AUC value is 85.45, which is comparable to the highest AUC generated by XGBoost at 85.73.

Table 4: AUC measurements using EW, REG, LR, and MI rescalors applied to the original, quadratic, and cubic feature sets in the GMC dataset's testing set when clustering only positive cases. The benchmark AUCs for XGBoost and Logistic Regression, derived when $K = 1$ and utilizing the EW rescalor, are underlined. The highest AUC values for both XGBoost and Logistic Regression are highlighted in bold. For Logistic Regression, a plus denotes a configuration that yield the highest AUC.

Í	XGBoost				Logistic Regression			
K	EW	REG	LR	MI	EW	REG	LR	MI
The original feature set								
1	<u>85.66</u>	**85.73**	85.66	85.66	<u>83.17</u>	84.02	83.19	83.48
2	85.43	84.98	85.08	85.03	83.65	83.15	83.57	83.44
3	84.65	84.48	84.99	84.53	84.08	82.88	84.61	83.19
4	83.84	84.30	84.43	83.86	84.20	82.92	84.56	82.92
The quadratic feature set								
1	85.35	85.24	85.24	85.23	84.36	84.43	84.51	84.46
2	84.13	84.59	84.49	84.47	84.70	83.96	85.43	84.09
3	83.27	83.95	84.07	84.08	84.69	84.03	85.53	84.46
4	83.32	83.44	83.61	83.45	84.36	84.00	85.39	83.87
The cubic feature set								
1	84.87	84.87	84.77	84.87	83.44	84.72	85.26	82.51
2	84.05	84.17	83.78	84.24	83.00	84.66	85.27	82.74
3	83.59	83.24	83.15	83.49	83.14	84.73	85.39	82.55
4	82.65	82.85	82.60	82.86	83.17	84.84	**85.45**$^{+}$	82.60

5. Conclusion

In this paper, we introduced an innovative method, termed as "rescaled cluster-then-predict," which employs a two-step process: first, the features are rescaled according to information retrieved from both the features and target variable, followed by the creation of individual models after clustering data into specific subgroups.

Through the application of this methodology on two credit-scoring datasets, we demonstrated that clustering rescaled quadratic or cubic features enhances the performance of Logistic Regression, outperforming XGBoost in terms of AUC in the testing set. This points to the generalizability of our approach across various applications.

Moreover, this study offers a method for clustering only positive cases, proving it to be as effective as all-case clustering while providing superior efficiency. Future work can explore the application of the "rescaled cluster-then-predict" strategy across other domains and investigate the development of more sophisticated rescaling techniques to further refine the process.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used ChatGPT in order to improve language and readability. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Funding

The authors gratefully acknowledge the financial support they received for this work. The National Yang Ming Chiao Tung University's Higher Education Sprout Project, which was funded by the Ministry of Education of Taiwan, provided essential resources. Additionally, the Ministry of Science and Technology of Taiwan significantly contributed to the funding through Grants 110-2115-M-A49-010-MY2, 111-2118-M-A49-007, and 112-2118-M-A49-001-MY2.

References

Aggarwal, C.C., 2015. Data mining: The textbook. Springer.

Arrieta, A.B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., et al., 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible ai. Information Fusion 58, 82–115.

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

Brundage, M., Avin, S., Clark, J., Toner, H., Eckersley, P., Garfinkel, B., Dafoe, A., Scharre, P., Zeitzoff, T., Filar, B., et al., 2018. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. Technical Report. arXiv preprint arXiv:1802.07228.

Bussmann, N., Giudici, P., Marinelli, D., Papenbrock, J., 2020. Explainable ai in fintech risk management. Frontiers in Artificial Intelligence 3, 26.

Chang, Y.C., Chang, K.H., Wu, G.J., 2018. Application of extreme gradient boosting trees in the construction of credit risk assessment models for financial institutions. Applied Soft Computing 73, 914–920.

Chen, C., Lin, K., Rudin, C., Shaposhnik, Y., Wang, S., Wang, T., 2018. An interpretable model with globally consistent explanations for credit risk. arXiv preprint arXiv:1811.12615 .

Chen, C., Shyu, M.L., 2011. Clustering-based binary-class classification for imbalanced data sets, in: 2011 IEEE International Conference on Information Reuse & Integration, IEEE. pp. 384–389.

Choi, J., Kwon, H.J., 2015. The information filtering of gene network for chronic diseases: Social network perspective. International Journal of Distributed Sensor Networks 11, 736569.

Demajo, L.M., Vella, V., Dingli, A., 2020. Explainable AI for interpretable credit scoring. arXiv preprint arXiv:2012.03749 .

Fernández, A., García, S., Galar, M., Prati, R.C., Krawczyk, B., Herrera, F., 2018. Learning from imbalanced data sets. volume 10. Springer.

Freund, Y., Schapire, R., Abe, N., 1999. A short introduction to boosting. Journal-Japanese Society For Artificial Intelligence 14, 1612.

Friedman, J.H., 2001. Greedy function approximation: A gradient boosting machine. Annals of statistics 29, 1189–1232.

García, S., Luengo, J., Herrera, F., 2015. Data preprocessing in data mining. Springer.

Géron, A., 2022. Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow. O'Reilly Media, Inc.

Goodman, B., Flaxman, S., 2017. European union regulations on algorithmic decision-making and a "right to explanation". AI magazine 38, 50–57.

Gray, R.M., 2011. Entropy and information theory. Springer Science & Business Media.

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

Hastie, T., Tibshirani, R., Friedman, J.H., Friedman, J.H., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. volume 2. Springer.

He, H., Garcia, E.A., 2009. Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering 21, 1263–1284.

James, G., Witten, D., Hastie, T., Tibshirani, R., 2013. An Introduction to Statistical Learning. volume 112. Springer.

Krawczyk, B., 2016. Learning from imbalanced data: open challenges and future directions. Progress in Artificial Intelligence 5, 221–232.

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

Liu, W., Fan, H., Xia, M., 2022. Credit scoring based on tree-enhanced gradient boosting decision trees. Expert Systems with Applications 189, 116034.

Menard, S., 2002. Applied logistic regression analysis. 106, Sage.

Misheva, B.H., Osterrieder, J., Hirsa, A., Kulkarni, O., Lin, S.F., 2021. Explainable AI in credit risk management. arXiv preprint arXiv:2103.00949 .

Peikari, M., Salama, S., Nofech-Mozes, S., Martel, A.L., 2018. A cluster-then-label semi-supervised learning approach for pathology image classification. Scientific Reports 8, 1–13.

Qin, C., Zhang, Y., Bao, F., Zhang, C., Liu, P., Liu, P., 2021. Xgboost optimized by adaptive particle swarm optimization for credit scoring. Mathematical Problems in Engineering 2021, 1–18.

Qiu, W., 2019. Credit risk prediction in an imbalanced social lending environment based on xgboost, in: 2019 5th International Conference on Big Data and Information Analytics (BigDIA), IEEE. pp. 150–156.

Radu, V., Katsikouli, P., Sarkar, R., Marina, M.K., 2014. A semi-supervised learning approach for robust indoor-outdoor detection with smartphones, in: Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems, pp. 280–294.

Ross, B.C., 2014. Mutual information between discrete and continuous data sets. PloS One 9, e87357.

Schapire, R.E., 1999. A brief introduction to boosting, in: Ijcai, Citeseer. pp. 1401–1406.

Selbst, A., Powles, J., 2018. "meaningful information" and the right to explanation, in: Friedler, S.A., Wilson, C. (Eds.), Proceedings of the 1st Conference on Fairness, Accountability and Transparency, PMLR. pp. 48–48. URL: https://proceedings.mlr.press/v81/selbst18a.html.

Sun, Y., Wong, A.K., Kamel, M.S., 2009. Classification of imbalanced data: A review. International journal of pattern recognition and artificial intelligence 23, 687–719.

Thomas, L., Crook, J., Edelman, D., 2017. Credit scoring and its applications. SIAM.

Tsai, C.F., 2014. Combining cluster analysis with classifier ensembles to predict financial distress. Information Fusion 16, 46–58.

Xia, Y., Liu, C., Li, Y., Liu, N., 2017. A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring. Expert systems with applications 78, 225–241.

Appendix

Appendix A: The PAK dataset

The feature "ID" was removed in the data because it is simply a sequential number without meaningful impact. We remove features with only one unique value and remove features about geographic information. Table 5 is deferred to the Appendix to summarize the final preprocessed dataset, including the binary target y and the original features $x_1, \ldots, x_{12}$.

Table 5: Processed PAK dataset: A summary statistics overview

	mean	std	min	50%	max	skewness	kurtosis
y	0.3	0.4	0.0	0.0	1.0	1.1	-0.8
x_1	0.0	1.0	-1.8	-0.4	1.8	0.5	-0.6
x_2	0.0	1.0	-0.7	-0.7	1.9	0.7	-1.4
x_3	0.0	1.0	-0.9	-0.3	20.5	2.0	10.1
x_4	0.1	0.3	0.0	0.0	1.0	3.2	8.3
x_5	0.0	1.0	-3.7	-0.2	12.0	1.7	7.4
x_6	-0.0	1.0	-0.2	-0.2	5.7	3.9	13.3
x_7	-0.0	1.0	-0.7	-0.7	3.4	0.6	-1.6
x_8	-0.0	1.0	-0.7	-0.7	3.4	0.6	-1.6
x_9	0.0	1.0	-0.2	-0.2	5.9	4.3	16.4
x_{10}	0.0	1.0	-0.7	-0.7	1.4	0.7	-1.5
x_{11}	0.0	1.0	-0.0	-0.0	28.3	25.7	658.5
x_{12}	0.0	1.0	-2.5	-0.2	4.2	0.5	-0.4

Appendix B: The GMC dataset

For missing value imputation, features x_5 (MonIncome) and x_{11} (NOfDependents) have missing data ratios of 19.8% and 2.6%, respectively. We add two new features, x_6 and x_{12}, to the dataset to indicate missing values in x_5 and x_{11}, respectively. After preprocessing, the final dataset has 12 features. Table 6 shows the summary statistics of the target (y) and 12 features ($x_1, \ldots, x_{12}$).

Table 6: Processed GMC dataset: A summary statistics overview

	mean	std	min	50%	max	skewness	kurtosis
y	0.1	0.2	0.0	0.0	1.0	3.5	10.0
x_1	-0.0	1.0	-3.1	0.3	3.9	-2.1	4.2
x_2	-0.0	1.0	-3.5	-0.0	3.8	0.2	-0.5
x_3	0.0	1.0	-0.4	-0.4	3.0	1.9	1.5
x_4	0.0	1.0	-2.4	-0.2	3.3	0.7	0.6
x_5	-0.0	0.9	-7.6	0.0	3.3	-6.9	54.4
x_6	0.2	0.4	0.0	0.0	1.0	1.5	0.3
x_7	-0.0	1.0	-1.6	-0.1	9.6	1.2	3.1
x_8	0.0	1.0	-0.2	-0.2	5.2	3.9	13.3
x_9	-0.0	1.0	-1.3	0.7	1.3	-0.5	-1.7
x_{10}	0.0	1.0	-0.2	-0.2	5.5	4.1	15.1
x_{11}	0.0	1.0	-0.7	-0.7	17.3	1.6	3.2
x_{12}	0.0	0.2	0.0	0.0	1.0	5.9	33.3