Proceedings of the 4th International Conference on Machine Learning and Automation
DOI: 10.54254/2755-2721/2026.37463

A Comparison of AutoML Platforms for Loan Default Prediction: An Empirical Study of H2O AutoML, TPOT, and Azure AutoML

Zhicheng Lin

School of Statistics and Applied Mathematics, Anhui University of Finance and Economics, Bengbu, China
3394157865@qq.com

Abstract. With the development of digital finance and the expansion of credit businesses, loan default prediction has become increasingly important in financial risk management. Traditional machine learning typically requires manual model selection and repeated parameter tuning, which demands substantial modeling experience. This study used the Credit Risk Dataset to compare the performance of H2O AutoML, TPOT, and Azure AutoML on the loan default prediction task, and evaluated them using AUC, Accuracy, Precision, Recall, and F1-score. The results showed that, under each platform's native validation procedure, H2O AutoML achieved relatively high results on most metrics, with AUC, Accuracy, and F1-score of 0.918, 0.902, and 0.724, respectively; Azure AutoML's AUC_weighted, Accuracy, and F1-score were 0.917, 0.899, and 0.718, respectively, which were close to those of H2O AutoML; TPOT attained the highest Precision at 0.948, but its Recall was 0.524. The findings indicated that different AutoML platforms exhibited varying performance in loan default prediction, and platform selection should be aligned with specific task requirements.

Keywords: Automated Machine Learning (AutoML), Loan Default Prediction, Credit Risk, Class Imbalance, Model Comparison, Hyperparameter Optimization

1. Introduction

With the growth of digital finance and the expansion of credit business, lending services have become more widespread, and risk management has gradually emerged as an increasingly prominent topic. Inaccurate prediction of credit loan default risk may expose financial institutions to substantial bad debts and financial losses. In practical terms, improving the accuracy of loan default risk prediction helps financial institutions better identify high-risk borrowers, thereby reducing bad debts and financial losses. Machine learning is commonly used for classification and prediction tasks [1] and has been applied to consumer credit risk prediction [2]. However, traditional modeling typically requires manual model selection, feature processing, and parameter tuning, while hyperparameter settings directly affect model training and prediction outcomes [3]. For users lacking extensive modeling experience, this process is complex and time-consuming. The advent of AutoML offers a

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2026.37463

new approach to this problem by automatically performing steps such as model selection and parameter optimization [4].

This paper addressed the loan default risk prediction problem from the perspective of comparing the predictive performance of different AutoML platforms. Loan default prediction can be viewed as a binary classification problem, that is, determining whether a borrower is likely to default based on basic borrower characteristics, loan-related variables, and credit history variables. Although traditional machine learning methods can be used for this task, their modeling process typically requires manual model selection, feature processing, and parameter tuning. By contrast, the defining feature of AutoML is the automation of model selection, parameter optimization, and model comparison [4]; it is therefore well suited to analyzing loan default risk prediction.

This paper is organized into five parts. The second part reviews the literature on credit risk prediction and AutoML. The third part introduces the research data, methodology, and evaluation metrics. The fourth part compares and analyzes the experimental results of H2O AutoML, TPOT, and Azure AutoML. Finally, the fifth part concludes the paper and discusses its limitations and directions for future research.

2. Literature review

Machine learning is commonly used for classification, prediction, and pattern recognition, where the classification task in supervised learning mainly trains a model on labeled data and then determines the class to which a new sample belongs [1]. In the credit risk domain, Khandani et al. [2] used machine learning to build a consumer credit risk prediction model, demonstrating that machine learning can be applied to default risk identification. Traditional machine learning modeling still requires manual handling of model selection, features, and parameters. Wu et al. [3] pointed out that hyperparameters affect model training and prediction outcomes, so tuning often requires domain expertise and repeated experimentation. Kotthoff et al. [4] integrated algorithm selection and hyperparameter optimization in Auto-WEKA 2.0, completing model configuration through automatic search. In this paper, AutoML mainly refers to automated steps such as model selection, hyperparameter optimization, and model comparison, aimed at reducing manual modeling effort in loan default prediction.

As AutoML tools continue to evolve, their capabilities have gradually expanded from automatic model selection and parameter optimization to more complete machine learning pipelines. Truong et al. [5] noted that different AutoML tools differ in their support for data processing, model search, and result analysis. For loan default prediction, Xu et al. [6] applied multiple machine learning methods to default prediction in China's P2P lending market. Shi et al. [7] further provided a systematic review of machine learning-driven credit risk research. In recent years, Bussmann et al. [8] and Bücker et al. [9] focused on model interpretability, transparency, and auditability from the perspectives of credit risk management and credit scoring. Barbudo et al. [10] summarized the research categories and development trends of AutoML, while Baratchi et al. [11] reviewed the field's past, present, and future. Yang et al. [12] further applied AutoML to credit scoring classification, showing that automated modeling has entered concrete credit risk applications. Overall, related research has extended from automated modeling itself to platform differences, model interpretation, and practical applications; this paper therefore compared the performance of H2O AutoML, TPOT, and Azure AutoML on the same loan default prediction dataset.

DOI: 10.54254/2755-2721/2026.37463

3. Research methods

3.1. Research framework

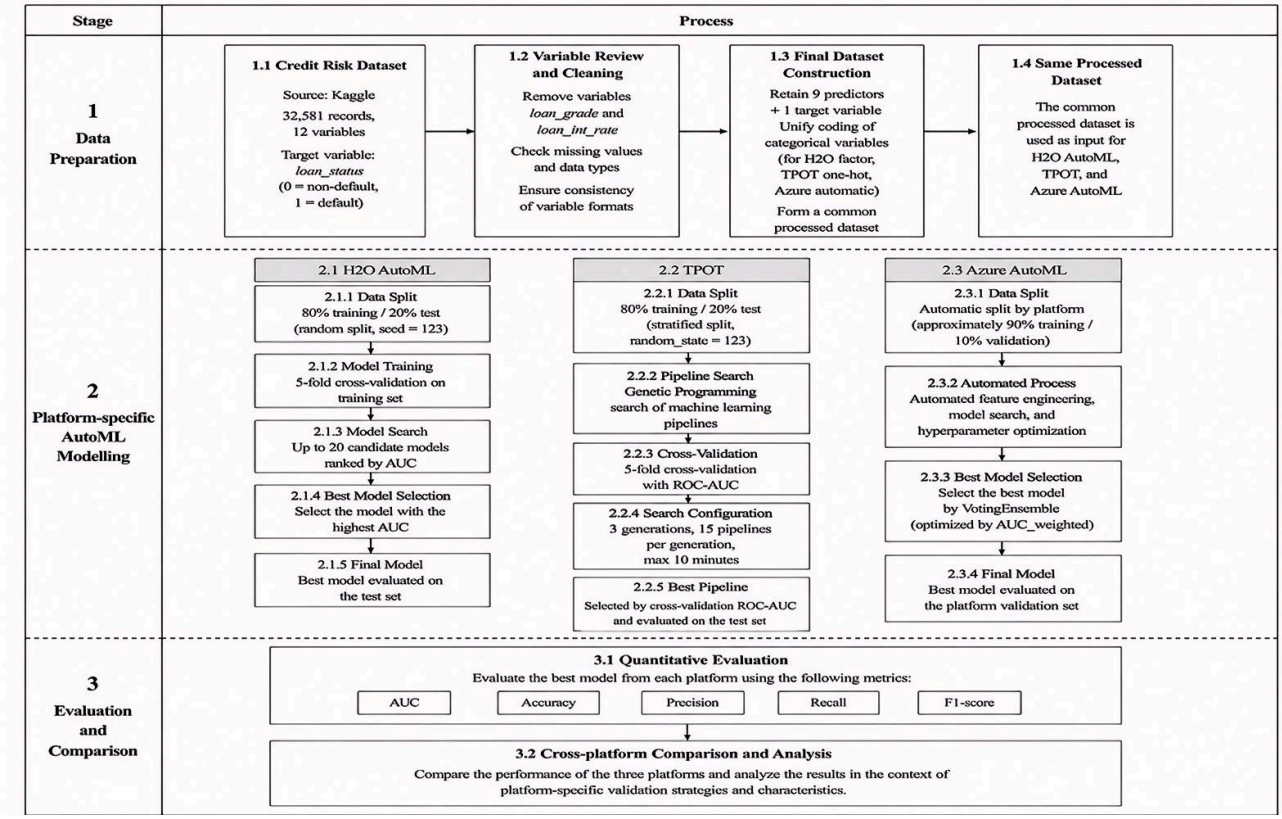


Figure 1. Research framework and experimental workflow

Figure 1 summarizes the complete workflow of this study, from raw data processing to cross-platform result comparison, which consists of three stages: data preparation, platform-specific AutoML modeling, and evaluation and comparison. To control for differences in input data, all three platforms were based on the same reviewed and cleaned dataset; at the same time, each platform retained its own training, validation, and model search approach to reflect its actual usage workflow.

The first stage was data preparation. The study first used the Credit Risk Dataset, which contains 32,581 records and 12 variables, and set loan_status as the binary target variable, where 0 indicates non-default and 1 indicates default. The meaning, data type, and missingness of each variable were then examined, and loan_grade and loan_int_rate, which may contain preexisting risk ratings and pricing judgments, were removed. The processed data retained 9 predictor variables and 1 target variable. The three platforms used the same data source and retained variables, but the categorical variables were handled according to platform requirements: H2O's factor type, TPOT's one-hot encoding, and Azure AutoML's automated feature processing.

The second stage was platform-specific AutoML modeling. H2O AutoML randomly split the data into training and test sets at 80% and 20%, and performed 5-fold cross-validation within the training set; the platform trained up to 20 candidate models, ranked them by AUC, and selected the best model, with the test set used only for final evaluation. TPOT adopted an 80%/20% stratified split and searched machine learning pipelines via genetic programming, using 5-fold cross-validated ROC-AUC as the search criterion, and selected the best pipeline under settings of 3 generations, 15

DOI: 10.54254/2755-2721/2026.37463

candidate pipelines per generation, and a maximum of 10 minutes. Azure AutoML did not have the researcher manually specify an independent validation set; instead, it used a platform-managed split of about 90% training and 10% validation, and automatically performed featureization, model search, and hyperparameter optimization, finally selecting the best model based on AUC_weighted.

The third stage was evaluation and comparison. The study extracted the best model or pipeline selected by each platform and described predictive performance using AUC, Accuracy, Precision, Recall, and F1-score. Among these, AUC measures the model's overall ability to distinguish defaulting from non-defaulting samples, Accuracy reflects the overall prediction correctness, Precision and Recall represent the reliability of default prediction and the identification of actual default samples, respectively, and F1-score comprehensively balances Precision and Recall. Because Azure AutoML used an internal validation set while H2O AutoML and TPOT used their respective test sets, this study conducted a cross-platform analysis combining metric results with the validation procedures, and the conclusions served to illustrate the empirical performance under each platform's native workflow rather than a strict algorithmic ranking under identical experimental conditions.

In addition, to provide a reference for non-AutoML methods, this paper separately constructed a manual logistic regression baseline. Because this baseline did not involve automatic model search, Figure 1 did not include it in the platform-specific AutoML modeling workflow; its data processing, model settings, and evaluation results were reported separately in Part 4.

3.2. Dataset selection and data preparation

3.2.1. Dataset selection

This study used the Credit Risk Dataset from Kaggle [13], which contains 32,581 records and 12 variables, covering basic borrower characteristics, loan information, and credit history. The target variable loan_status is a binary variable, where 0 indicates non-default and 1 indicates default. The data include both numerical and categorical variables for modeling and comparison across the three AutoML platforms.

3.2.2. Dataset variables

After removing loan_grade and loan_int_rate, the final modeling data contains 9 predictor variables and 1 target variable; the variable definitions are given in Table 1.

Table 1. Final modeling variables and their definitions

Variable	Type	Description	Example
person_age	Numerical	Age of borrower	32
person_income	Numerical	Annual income	70,000
person_home_ownership	Categorical	Home ownership status	RENT
person_emp_length	Numerical	Employment length (years)	6
loan_intent	Categorical	Loan purpose	MEDICAL
loan_amnt	Numerical	Loan amount	13,200

DOI: 10.54254/2755-2721/2026.37463

Table 1. (continued)

loan_status	Binary target	0 = non-default; 1 = default	0 (non-default)
loan_percent_income	Numerical	Loan amount / annual income	0.19
cb_person_default_on_file	Categorical	Historical default record	N
cb_person_cred_hist_length	Numerical	Credit history length	6

Note: The example values are taken from the same complete record in the original dataset.

The predictor variables mainly include three categories: basic borrower characteristics, loan information, and credit history; for the target variable loan_status, 0 indicates non-default and 1 indicates default.

3.2.3. Data preparation

Data preprocessing is a key step in feature engineering. Before modeling, loan_grade and loan_int_rate were removed to reduce the model's reliance on preexisting risk ratings and pricing information.

The three platforms used the same processed data but differed in preprocessing: H2O set the 3 categorical variables as factors; TPOT one-hot encoded the categorical variables and imputed numerical missing values with the training-set median; Azure performed subsequent featureization automatically via the platform.

For data splitting, H2O used an 80% training set and 20% test set (seed=123); TPOT used an 80%/20% stratified split (random_state=123); Azure did not separately specify a validation or test set and adopted the platform default of about 90% training and 10% validation.

Regarding the target variable distribution, the original data contained 25,473 non-default samples (78.18%) and 7,108 default samples (21.82%). This distribution indicated a clear class imbalance; therefore, besides Accuracy, the subsequent evaluation focused on AUC, Recall, and F1-score.

3.3. Automl platforms and modelling process

3.3.1. H2O automl

H2O AutoML was invoked via R and ran on Java. The data were split 80%/20%, 5-fold cross-validation was performed within the training set, up to 20 candidate models were trained, ranking was by AUC with class balancing enabled, and the test set was used only for final evaluation.

3.3.2. TPOT

TPOT is based on Python and uses genetic programming to search complete machine learning pipelines. The data were split 80%/20% stratified, 5-fold cross-validation was used, and candidate pipelines were evaluated by ROC-AUC; the search settings were 3 generations with 15 candidate pipelines per generation and a maximum of 10 minutes.

DOI: 10.54254/2755-2721/2026.37463

3.3.3. Azure automl

Azure AutoML is a cloud-based automated machine learning platform. This study uploaded the processed data and set loan_status as the classification target, and the platform automatically performed featureization, model search, and parameter optimization; no independent validation set, test set, or cross-validation was specified, the default split of about 90% training and 10% validation was used, and AUC_weighted was the primary optimization metric.

3.3.4. Main model families

The candidate models of the three platforms mainly included Logistic Regression/GLM, Random Forest/Extra Trees, GBM/XGBoost/LightGBM, Deep Learning, and Ensemble/Stacking. The first four types are used for classification modeling, while Ensemble/Stacking is used to combine the predictions of multiple base models.

3.3.5. Platform selection

In the early stage of the study, Google Cloud AutoML and Baidu EasyDL were also attempted, but the former did not yield complete results due to pipeline errors and the latter was not continued because of additional computing costs. Therefore, the final comparison was conducted among H2O AutoML, TPOT, and Azure AutoML.

3.4. Model evaluation methods

The prediction task in this paper was binary classification, with loan_status=1 denoting default and serving as the positive class. Based on the actual experimental outputs, this paper evaluated model performance using AUC, Accuracy, Precision, Recall, and F1-score.

3.4.1. Confusion matrix

In the confusion matrix, TN, FP, FN, and TP denote true negatives, false positives, false negatives, and true positives, respectively. The specific statistics of each model on the final test set or validation set are given in Table 2.

Table 2. Confusion matrix statistics of each model

Model	TN	FP	FN	TP	Total (N)
H2O AutoML	4,975	87	546	829	6,437
TPOT	5,054	41	677	745	6,517
Azure AutoML	2,509	39	291	420	3,259
Logistic Regression (baseline)	4,882	213	869	553	6,517

Note: H2O used probabilistic splitting with a target training-to-test ratio of 80:20, yielding 6,437 test observations. TPOT and logistic regression each used 6,517 test observations from an 80:20 stratified split. Azure AutoML used an internal validation set containing 3,259 observations. The evaluation sets differed in size and composition; therefore, the results do not constitute a comparison on a common test set.

DOI: 10.54254/2755-2721/2026.37463

3.4.2. Classification metrics

Accuracy is the proportion of correctly predicted samples among all samples, reflecting the overall classification correctness of the model.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1}$$

Precision is the proportion of actually defaulting samples among all samples predicted as default, measuring the reliability of default prediction.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

Recall is the proportion of actual default samples correctly identified by the model, measuring the model's ability to detect default samples.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

F1-score is the harmonic mean of Precision and Recall, used to comprehensively evaluate the balance between the two.

$$\text{F1} - \text{score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

AUC is the area under the ROC curve. The ROC curve reflects the relationship between the true positive rate (TPR) and the false positive rate (FPR) at different classification thresholds.

$$\text{TPR} = \frac{\text{TP}}{\text{TP} + \text{FN}} \, \text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}} \tag{5}$$

$$\text{AUC} = \int_0^1 \text{TPR}(\text{FPR}) \, \text{d}(\text{FPR}) \tag{6}$$

Because the dataset contained more non-default than default samples, using Accuracy alone may not fully reflect the model's ability to identify default samples. Therefore, this paper took AUC as the primary reference metric and combined Accuracy, Precision, Recall, and F1-score for a comprehensive comparison of model performance.

4. Experiments

4.1. Experimental setup

This paper used H2O AutoML, TPOT, and Azure AutoML to conduct loan default prediction experiments, and additionally constructed a manual logistic regression as a traditional baseline. H2O AutoML, TPOT, and logistic regression ran on a local computer with an Intel Core i9-14900HX processor, 32 GB memory, and an NVIDIA GeForce RTX 4070 Laptop GPU, under 64-bit Windows 11; Azure AutoML ran in a Serverless cloud computing environment.

H2O AutoML used R 4.5.2, H2O 3.44.0.3, and OpenJDK 17.0.19; TPOT used Python 3.10.20, TPOT 0.12.2, scikit-learn 1.5.2, and XGBoost 3.2.0; Azure AutoML used Serverless computing with the Standard_DS3_v2 compute specification; the logistic regression baseline used Python 3.10.20

DOI: 10.54254/2755-2721/2026.37463

and scikit-learn 1.5.2. The search, validation, and time limits of each experiment are given in Table 3.

To ensure the reproducibility of the baseline results, logistic regression used the same processed data as TPOT and adopted an 80% training and 20% test stratified split (random_state=123). Numerical variables were imputed with the training-set median and standardized, and categorical variables were one-hot encoded; the model used the liblinear solver with a classification threshold of 0.5.

Table 3. Experimental environment and main settings of the AutoML platforms and the logistic regression baseline

Platform	Environment	Software / version	Main setting	Time limit
H2O AutoML	Local	R 4.5.2 H2O 3.44.0.3 OpenJDK 17.0.19	Candidates: 20 models Validation: 5-fold CV Metric: AUC	No explicit limit
TPOT	Local	Python 3.10.20 TPOT 0.12.2 scikit-learn 1.5.2 XGBoost 3.2.0	Candidates: 15 pipelines × 3 generations Validation: 5-fold CV Metric: ROC-AUC	10 min
Azure AutoML	Serverless	Azure AutoML (managed service) Standard_DS3_v2	Candidates: platform-managed search Validation: ~10% internal validation Metric: AUC_weighted	360 min
Logistic Regression (baseline)	Local	Python 3.10.20 scikit-learn 1.5.2	Model: manually specified logistic regression Evaluation: 20% hold-out test set Metric: AUC	No explicit limit

4.2. Experimental results

All three AutoML platforms yielded valid classification results. The final best model of H2O AutoML was StackedEnsemble_AllModels, the best pipeline of TPOT consisted of XGBClassifier and GradientBoostingClassifier, and the best model of Azure AutoML was VotingEnsemble. In addition, this paper included a manual logistic regression as a simple traditional baseline to illustrate the performance difference of the AutoML pipeline relative to a single preset model. The main evaluation metrics of each method are given in Table 4.

Table 4. Model performance comparison of AutoML platforms and the logistic regression baseline

Platform	AUC ↑	Accuracy ↑	Precision ↑	Recall ↑	F1-score ↑
H2O AutoML	0.918	0.902	0.905	0.603	0.724
TPOT	0.907	0.890	0.948	0.524	0.675
Azure AutoML	0.917*	0.899	0.915	0.591	0.718

DOI: 10.54254/2755-2721/2026.37463

Table 4. (continued)

Logistic Regression† (baseline)	0.802	0.834	0.722	0.389	0.505

Note: * Azure AutoML reports AUC_weighted on the internal validation set; H2O AutoML and TPOT report AUC on their respective test sets. Azure's Precision, Recall, and F1-score are recalculated from the confusion matrix with the default class (1) as positive. † The logistic regression baseline uses the same processed data and an 80%/20% stratified split (random_state=123); its metrics come from an independent test set.

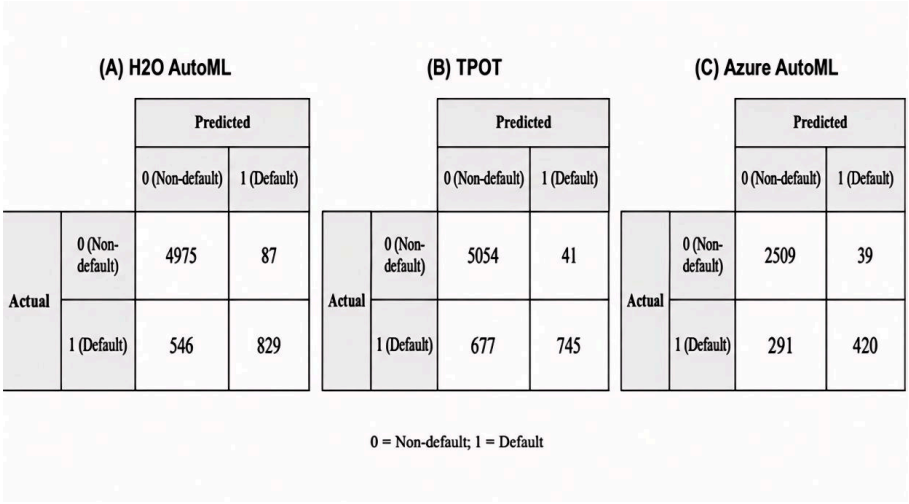


Figure 2. Confusion matrices of the three AutoML platforms

Figure 2 shows the confusion matrices of the three platforms. Because Azure AutoML used an internal validation set while H2O AutoML and TPOT used independent test sets, the three matrices had different total sample sizes; therefore, absolute counts were not compared directly, and the following discussion mainly compared ratio-type metrics such as AUC, Accuracy, Precision, Recall, and F1-score.

4.3. Results comparison

H2O AutoML achieved relatively high results on AUC, Accuracy, Recall, and F1-score, at 0.918, 0.902, 0.603, and 0.724, respectively. Azure AutoML's AUC_weighted, Accuracy, Recall, and F1-score were 0.917, 0.899, 0.591, and 0.718, respectively, close to those of H2O AutoML. TPOT's Precision was 0.948, the highest among the methods, but its Recall was 0.524 and F1-score was 0.675.

TPOT's higher Precision indicated that its default prediction was more conservative, but its lower Recall showed that many actual default samples remained unidentified. In contrast, H2O AutoML had slightly lower Precision but higher Recall and F1-score, performing a better balance between Precision and Recall. The manual logistic regression baseline had AUC, Accuracy, Precision, Recall, and F1-score of 0.802, 0.834, 0.722, 0.389, and 0.505, respectively, lower than the corresponding results reported by the three AutoML platforms. This baseline used the same 80%/20% stratified split as TPOT and mainly served as a reference for a simple traditional model.

DOI: 10.54254/2755-2721/2026.37463

4.4. Evaluation of experimental results

In loan default prediction, a False Negative denotes a sample that actually defaults but is predicted as non-default; such errors may leave high-risk borrowers unidentified in time. Therefore, besides Accuracy, Recall and F1-score are also important references. The results of this paper showed that H2O AutoML was relatively balanced on most metrics; TPOT had higher Precision but a lower identification rate of default samples; and Azure AutoML's overall results were close to those of H2O AutoML. The lower Recall and F1-score of the logistic regression baseline also indicated that a single simple model had limited default identification ability on the current class-imbalanced data.

At the same time, the validation mechanisms and search resources of the three platforms were not entirely consistent. H2O AutoML and TPOT used their respective 20% test sets for final evaluation, while Azure AutoML used the platform's internal validation set and reported AUC_weighted. Therefore, the results of this paper were mainly used to compare the actual performance of the three AutoML platforms under their respective native workflows, rather than a strict algorithmic ranking under identical conditions. In addition, this study used only a single credit risk dataset, and the results still needed further validation on more datasets.

5. Conclusion

This paper compared the performance of H2O AutoML, TPOT, and Azure AutoML in loan default prediction on the Kaggle Credit Risk Dataset, with a manual logistic regression added as a traditional baseline. The results showed that all three AutoML platforms achieved valid results under their respective native workflows: H2O AutoML was relatively balanced on most metrics, TPOT had the highest Precision, and Azure AutoML was close to H2O; the logistic regression baseline had relatively lower overall metrics. Limited by the differences in the platforms' default validation flows, the conclusions of this paper mainly reflected an empirical comparison under native flows rather than a strict ranking under fully identical conditions. This study still had the following limitations: only a single dataset was used; the cross-platform validation protocols were not unified; the metrics of different methods came from different test or validation sets; and the overall low Recall reflected that the class imbalance problem remained insufficiently addressed. Future work could rerun the methods under a unified data split, introduce class weights, SMOTE, or threshold calibration to improve Recall, add model interpretability analyses such as SHAP/LIME, and validate the generalization of the results on more datasets.

References

[1] Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering, 160*(1), 3–24.

[2] Khandani, A. E., Kim, A. J., & Lo, A. W. (2010). Consumer credit-risk models via machine-learning algorithms. *Journal of Banking & Finance, 34*(11), 2767–2787.

[3] Wu, J., et al. (2019). Hyperparameter optimization for machine learning models based on Bayesian optimization. *Journal of Electronic Science and Technology, 17*(1), 26–40.

[4] Kotthoff, L., et al. (2017). Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA. *Journal of Machine Learning Research, 18*(25), 1–5.

[5] Truong, A., et al. (2019). Towards automated machine learning: Evaluation and comparison of AutoML approaches and tools. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE.

[6] Xu, J., Lu, Z., & Xie, Y. (2021). Loan default prediction of Chinese P2P market: A machine learning methodology. *Scientific Reports, 11*(1), 18759.

DOI: 10.54254/2755-2721/2026.37463

[7] Shi, S., et al. (2022). Machine learning-driven credit risk: A systemic review. *Neural Computing and Applications, 34*(17), 14327–14339.
[8] Bussmann, N., et al. (2021). Explainable machine learning in credit risk management: N. Bussmann et al. *Computational Economics, 57*(1), 203–216.
[9] Bücker, M., et al. (2022). Transparency, auditability, and explainability of machine learning models in credit scoring. *Journal of the Operational Research Society, 73*(1), 70–90.
[10] Barbudo, R., Ventura, S., & Romero, J. R. (2023). Eight years of AutoML: Categorisation, review and trends. *Knowledge and Information Systems, 65*(12), 5097–5149.
[11] Baratchi, M., et al. (2024). Automated machine learning: Past, present and future: M. Baratchi et al. *Artificial Intelligence Review, 57*(5), 122.
[12] Yang, F., et al. (2025). BACS: Blockchain and AutoML-based technology for efficient credit scoring classification. *Annals of Operations Research, 345*(2), 703–723.
[13] Lao Tse. (n.d.). Credit risk dataset. *Kaggle*. www.kaggle.com/datasets/laotse/credit-risk-dataset. Accessed 15 Sept. 2026.