

Predictive Credit Risk Assessment

Mark Axelus

Faculty of Engineering & Computer
Science
University of Victoria
Victoria, BC
V01004718

Manya Goel

Faculty of Engineering & Computer
Science
University of Victoria
Victoria, BC
V01063457

Gerry Peng

Faculty of Engineering & Computer
Science
University of Victoria
Victoria, BC
V00944715

Christina McNeice

Faculty of Engineering & Computer
Science
University of Victoria
Victoria, BC
V00920553

Abstract

This study investigates whether machine learning models trained solely on readily available financial data can outperform traditional statistical approaches in predicting borrower default within two years. Using a dataset of over 16,000 credit records from 2011 to 2014, we evaluate three models: logistic regression, neural networks, and k-Nearest Neighbour (k-NN). Each model was developed using standardized feature engineering, hyperparameter tuning, and stratified cross-validation. Our results show that k-NN achieved the highest accuracy (76.6%), followed by neural networks (75.9%) and logistic regression baseline (71.85%). Compared to the baseline, k-NN improved accuracy by 4.75% while maintaining balanced precision and recall. Although logistic regression remains the most interpretable, k-NN offers case-based transparency and strong predictive performance. These findings demonstrate that machine learning models, especially k-NN, can enhance credit risk assessment using standard financial indicators, offering practical benefits to lending institutions.

1. Introduction and Problem Statement

The study aimed to build and compare machine learning models in the financial domain that could accurately predict borrower default risk, more efficiently and transparently than the traditional models banks have used. Credit underwriting leans heavily on FICO scorecards that do not adapt to changes in borrower behavior or macroeconomic changes. Due to this lack of adaptability, lenders either accept borrowers who are high-risk defaulters or reject credit-worthy low-risk applicants. Our research asks the following question:

“Can a machine learning model be trained on only readily available data and predict whether a borrower will default within the next two years - more accurately and transparently than the traditional statistical models that have been traditionally used by banks to determine borrower risks?”

2. Related Works

Credit risk assessment has evolved from traditional statistical methods to sophisticated machine learning approaches. Logistic regression remains foundational in credit scoring due to its interpretability and industry acceptance, with Costa e Silva et al. [1] achieving a high accuracy rate using Portuguese banking data (3,000+ customers, 10% default rate). Identifying six key predictors with interpretable odds ratios, however limited by the single-institution scope.

Recent benchmark studies establish systematic ML comparison methodologies for credit risk, particularly addressing unbalanced datasets [2]. Systematic reviews show deep learning models generally outperform classical algorithms in credit risk assessment [3], though with reduced interpretability. Contemporary studies demonstrate varied ML performance: ANN achieving $\sim 77.45\%$ accuracy and KNN $\sim 72.20\%$ [4], closely aligning with our results of 75.9% (Neural Network) and 76.6% (k-NN). Advanced ensemble methods like XGBoost achieve up to 99.4% accuracy [5], suggesting future improvement opportunities.

Our study addresses literature gaps by providing a direct comparison of traditional statistical methods, classic ML algorithms, and neural networks using identical pre-processing and evaluation frameworks.

3. Dataset

The study uses the Credit Risk Benchmark Dataset published by Adil Shamin on Kaggle, which contains 16,714 borrower records from 2011 to 2014. The dataset tracks borrower outcomes for two years, has 11 features, and a binary classification target. This dataset includes 11 featured variables covering real-world factors that determine loan decisions, such as the Financial health of the applicant, credit profile, payment history, and user demographic.

To clean the data for our application, we removed two duplicate records, capped extreme outliers at the 99th percentile, and checked for missing values. Capped extremes removed debt ratios that exceeded 60,000% as this would deeply skew the model results.

3.1 Feature Engineering Pipeline

The feature engineering process was developed iteratively in conjunction with model testing:

1. Initial feature standardization to ensure equal weighting in distance calculations
2. Logarithmic transformation of heavily skewed feature values
3. Feature importance evaluation through ablation studies
4. Dimensionality experiments to optimize the feature space

Each iteration of feature engineering was validated against model performance metrics, leading to the final selection of our feature set.

For clean and direct training, we combined correlated late payment features into `total_late` and `late_ratio`. Adjusted age binding to create meaningful age groups for analysis and performed log transformations and standardization. Log transformations convert extremely skewed distributions into bell-shaped curves, compressing wide value ranges into manageable and meaningful scales, making the data more symmetric and reducing outlier influence.

4. Our Models

4.1 Logistic Regression

4.1.1 Methodologies

For our baseline, we implemented a logistic regression model in order to try and make binary classifications as to whether or not a borrower would have a serious delinquency within the next two years. This was chosen because of its relative simplicity and straightforward application to this type of finance problem. Our implementation uses the sigmoid function to map our input features to a probability, which is how we make our binary classification.

4.1.2 Model Development Process

We began by creating a correlation matrix in order to get an idea of which features are highly correlated with one another and figure out the predictive value of each feature. After combining certain features due to their similarity and high correlation, we performed a logarithmic transformation and standardization to scale our data. Our features were chosen based on a mix of correlation and practical relevance to credit risk. We evaluated our model using 5-fold cross-validation in order to reduce the risk of overfitting due to the random subsets being used. This allowed us to examine and compare the accuracy, precision, recall, and F1 score at each fold.

4.1.3 Performance Evaluation

Although logistic regression was meant to be a simple baseline method, it achieved reasonably good results.

The final accuracy, precision, recall, and F1 score values are:

- **Accuracy:** 71.85%
- **Precision:** 72.04%
- **Recall:** 71.59%
- **F1-Score:** 71.70%

These values show a fairly good balance between the different metrics. The goal of our model is to classify individuals as to whether or not they make good borrowers. Since we are trying to help increase profit for the lender, we want to maximize our expected value by only making good loans. This means that neither false positives nor false negatives are desirable, but neither is more detrimental than the other. Additionally, our logistic regression model is easy to interpret and fast to train. Although its accuracy may not be as high as more complex models, it served as a good baseline for us to work from.

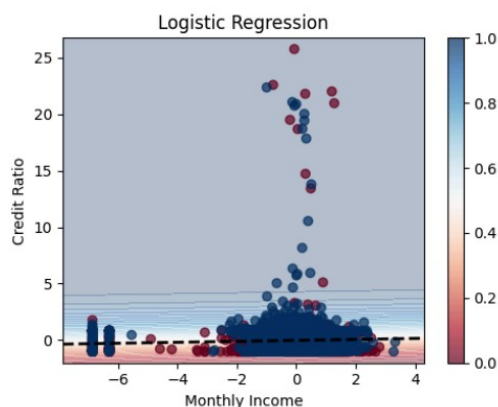


Figure 4.1.1: Decision Boundary

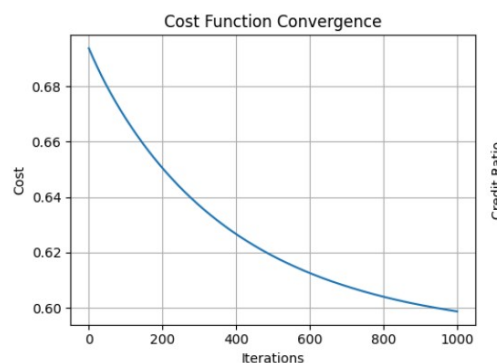


Figure 4.1.2: Cost Function Convergence

4.2 Neural Networks

4.2.1 Methodologies

We implemented a feedforward neural network to capture non-linear relationships between risk factors and default probability. Neural networks excel at automatic feature interaction detection and can achieve high predictive accuracy in credit risk applications. Our approach utilized forward and backward propagation with systematic hyperparameter optimization.

4.2.2 Model Development Process

Initial Model Analysis We began with a basic neural network architecture and generated learning curves to identify performance issues. The initial model exhibited high bias, indicating underfitting and insufficient model complexity to capture the underlying data patterns.

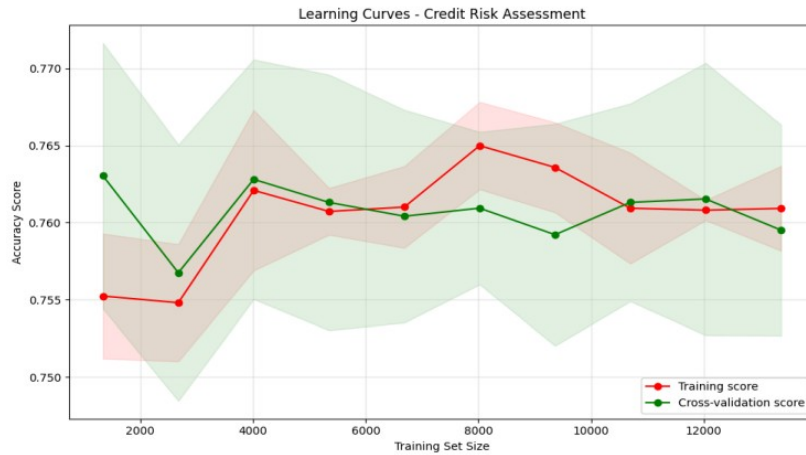


Figure 4.2.1: Learning curves for a basic [5,3,1] model

Architecture Optimization To address the high bias problem, we systematically modified the network architecture and performed cross-validation to identify optimal configurations. Multiple architectures were tested with varying hidden layer sizes and activation functions. After comparing different architectures, we select the [5,8,1] architecture with ReLU and sigmoid activation functions.

Hyperparameter Tuning The best-performing architecture underwent comprehensive hyperparameter optimization through cross-validation. Key parameters tuned included: learning rate, number of hidden layers, neurons per layer, and activation functions. Learning rates were systematically evaluated to balance convergence, speed, and model stability.

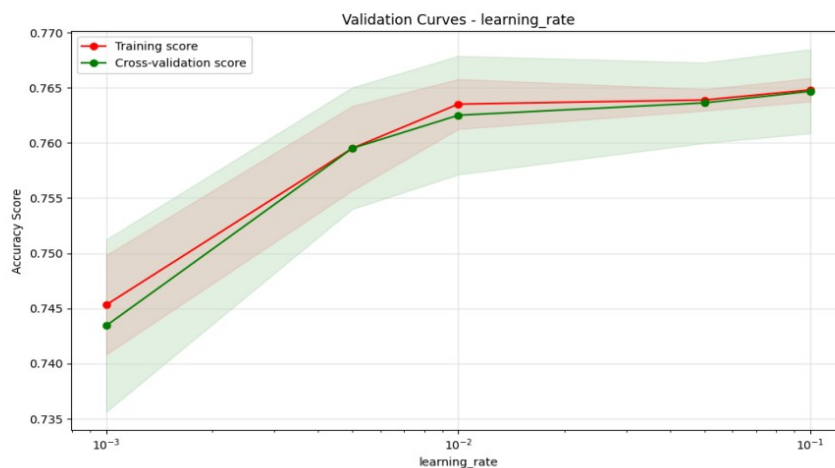


Figure 4.2.2: Learning Curves for hyperparameter tuning

4.2.3 Final Model Configuration

Our optimal neural network architecture consists of:

- **Structure:** [5, 8, 1] - 5 input features, 8 hidden neurons, 1 output neuron

- **Activation Functions:** ReLU for hidden layer, Sigmoid for output layer
- **Learning Rate:** 0.1000 (determined through systematic tuning)

4.2.4 Performance Evaluation

The final neural network model achieved:

- **Accuracy:** 75.86%
- **Precision:** 75.90%
- **Recall:** 75.86%
- **F1-Score:** 75.85%

These results represent a 4.01% improvement over the logistic regression baseline (71.85% accuracy), demonstrating the neural network's ability to capture non-linear patterns.

4.3 k-Nearest Neighbour (k-NN)

The k-NN algorithm presents a compelling approach to credit risk assessment due to its intuitive nature and interpretable results. While traditional credit scoring models often rely on parametric approaches, k-NN's non-parametric nature allows it to capture complex, non-linear relationships in credit data without making assumptions about the underlying distribution. This section explores the efficacy of k-NN in predicting credit defaults, with particular attention to the algorithm's behavior in feature space and its practical implications for risk assessment.

4.3.1 Methodologies

Our implementation of k-NN for credit risk assessment operates on the principle of similarity-based classification. Given a new credit application, the algorithm examines the k most similar historical credit records in the feature space to make a prediction. The similarity is computed using Euclidean distance in the standardized feature space, where each dimension represents a credit-relevant attribute such as payment history and credit utilization.

4.3.2 Feature Space Consideration

The implementation of k-NN for credit risk assessment demands careful attention to the structure and quality of the feature space. A central challenge is the curse of dimensionality, where the discriminative power of distance metrics such as Euclidean distance diminishes in high-dimensional spaces. As dimensionality increases, data points become sparser and less informative for nearest neighbor calculations, reducing model effectiveness.

This issue was empirically validated in our experiments, as shown in Fig 4.3.1, where model accuracy declined from approximately 73% to 61% as the number of features

increased to 100. The steep performance drop beyond 20 features underscores the need to reduce irrelevant or redundant variables.

To mitigate this effect, our approach prioritized feature quality over quantity. We conducted iterative testing and feature engineering, ultimately identifying two key predictors:

- Logarithmically transformed total late payments, and
- Revolving utilization

The logarithmic transformation was especially impactful, as it normalized the right-skewed distributions commonly found in financial data, allowing for more meaningful distance calculations within the k-NN framework.

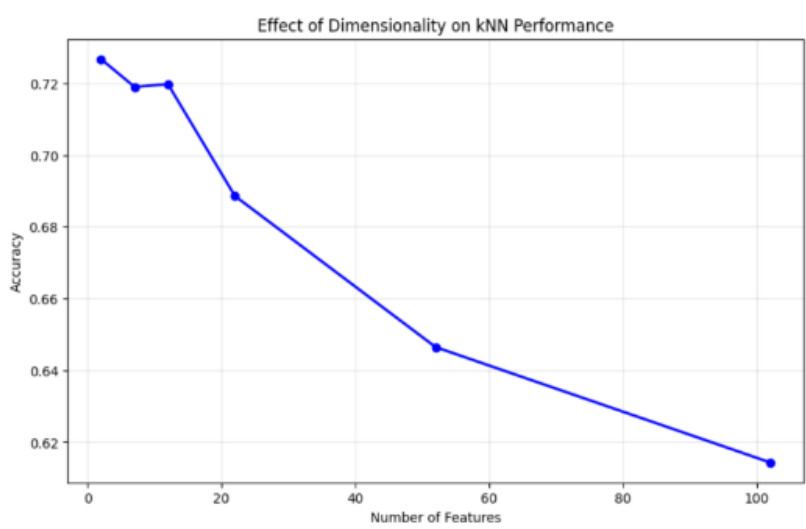


Figure 4.3.1: Curse of Dimensionality Graph

4.3.3 Model Development Process

Implementation Strategy Our k-NN implementation followed an iterative development approach, beginning with a basic model and progressively incorporating sophistication. The initial implementation utilized scikit-learn's `KNeighborsClassifier` as the foundation, which we then customized to accommodate our specific requirements for credit risk assessment.

Cross-Validation Framework To ensure reliable and unbiased model evaluation, we adopted a rigorous experimental design. First, the dataset was partitioned into an 80% training set and a 20% hold-out test set. All model development, including hyperparameter tuning and validation, was performed strictly on the training set to prevent data leakage and maintain the integrity of the final evaluation.

Within the training set, we implemented a stratified 5-fold cross-validation scheme. This approach preserved the class distribution of defaulters and non-defaulters across all folds, an essential consideration given the class imbalance typical in credit risk datasets.

By rotating through different training/validation splits, this framework provided robust estimates of model performance and helped mitigate overfitting during development.

4.3.4 Model Optimization Process

The optimization phase followed a systematic approach:

1. **Baseline Establishment:** We first established a baseline model using default parameters and basic feature preprocessing.
2. **Parameter Tuning:** A comprehensive grid search was conducted over the following parameters:
 - k-value range (1-60)
 - Distance metrics (Euclidean)
 - Weighting schemes (uniform vs distance-based)

Performance Monitoring: Each configuration was evaluated using:

- Cross-validation accuracy
 - Learning curve analysis
 - Error distribution patterns
 - Computational efficiency metrics
3. **Refinement Cycles:** Multiple refinement cycles were performed to:
 - Address identified performance bottlenecks
 - Optimize computational efficiency
 - Fine-tune feature preprocessing steps
 - Calibrate the final k-value selection

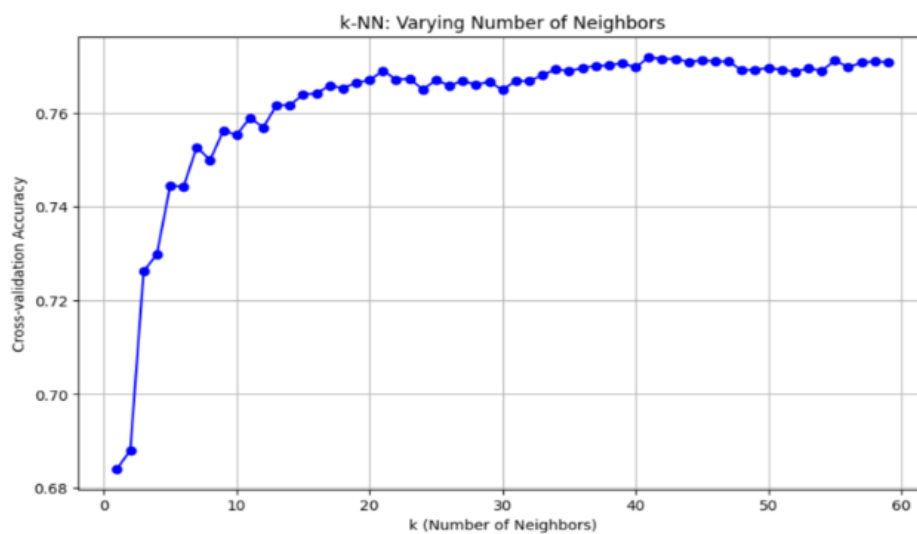


Figure 4.3.2: K-Fold Performance Chart

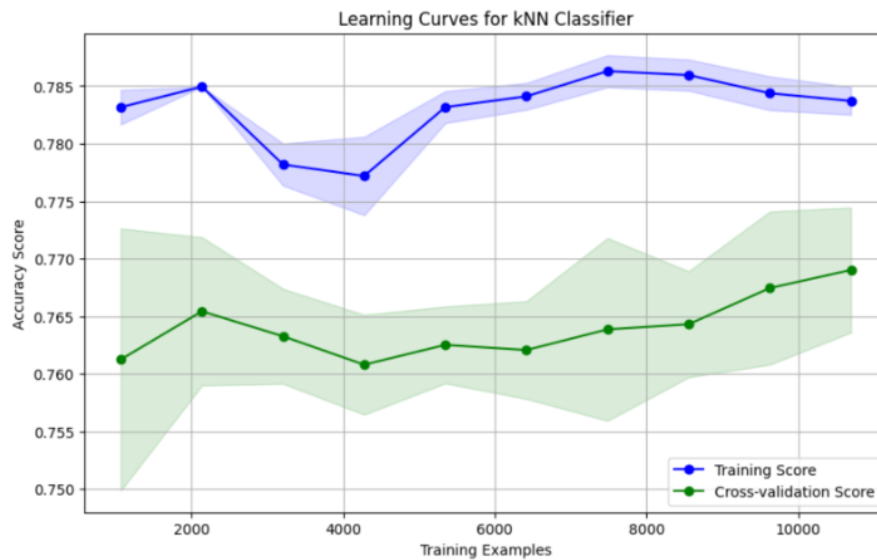


Figure 4.3.3: Learning Curve

Based on the results of our grid search, we selected $k = 21$ as the optimal number of neighbors. This value offered a strong balance between bias and variance, achieving a stable cross-validation accuracy (77%) while avoiding the noise sensitivity associated with smaller k values and the oversmoothing of larger ones. Furthermore, our choice of $k = 21$ is validated through the learning curve. We observe that adding more samples to the model may yield diminishing returns as it starts to plateau at 10,000 data samples. The tight confidence bands also show that the model performance is consistent throughout all splits of the data sample.

4.3.5 Performance Evaluation

The final optimized k-NN model achieved the following performance with $k = 21$ on the held-out test set of 20%:

- **Accuracy:** 76.60%
- **Precision:** 73.60%
- **Recall:** 75.50%
- **F1-Score:** 76.60%

These results represent a 4.75% improvement over the logistic regression baseline (71.85% accuracy), demonstrating the k-NN model's effectiveness. The performance highlights the algorithm's strength in leveraging feature similarity to capture localized, non-linear patterns within the credit risk data that are not as effectively modeled by a purely linear approach like logistic regression.

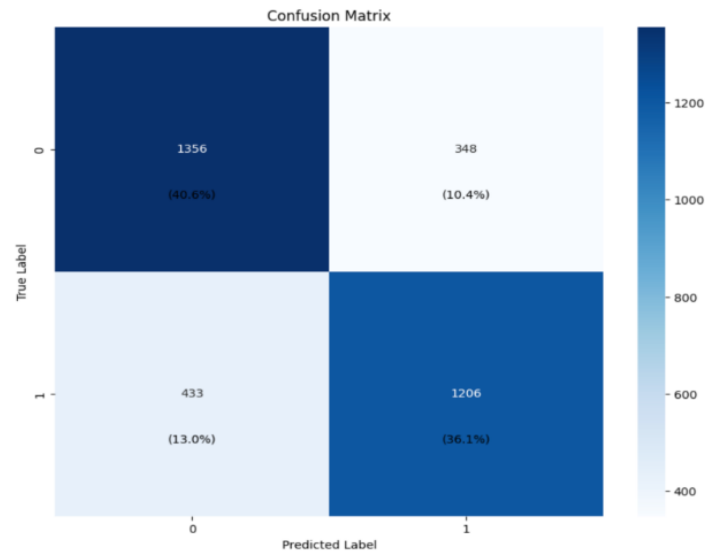


Figure 4.3.4: k-NN Confusion Matrix

The figure above presents the confusion matrix for the final k-NN model. The model shows relatively balanced predictive ability across both classes, but some misclassification remains, highlighting the trade-off between sensitivity and specificity in credit risk classification.

5. Statistical Analysis

5.1 Model Comparison and Performance Metrics

	Precision	Recall	Accuracy	F1-Score	Baseline Comparison
Logistic Regression	72.00%	71.50%	71.85%	71.7%	+0%
Neural Network	75.90%	75.86%	75.86%	75.85%	+4.01%
k-NN	73.60%	75.50%	76.60%	76.60%	+4.75%

Table 1: Model Performance Comparison

5.2 Validation

For model validation, we employed stratified k-fold cross-validation to preserve the class distribution throughout the validation process. Initially, cross-validation was used to address the high bias observed in our baseline neural network (Fig. 4.2.1). We systematically compared architectures with increased complexity, incorporating different activation function combinations and extended training epochs to improve model capacity.

Following architecture selection, we conducted a second round of cross-validation on the optimal model configuration to determine the best learning rate through hyperparameter tuning (Fig. 4.2.2). This two-stage validation approach ensured both architectural optimization and parameter fine-tuning while maintaining robust performance estimates.

6. Discussions

6.1 Model Limitations for Neural Network

Despite performance improvements, the neural network still exhibits high bias, suggesting potential for further architectural refinement. The model's "black box" nature presents interpretability challenges for practical financial applications, where regulatory requirements often demand explainable decision-making processes.

6.2 Model Limitations for k-NN

Due to the nature of kNN, it exhibits significant computational inefficiency by requiring the entire training dataset for predictions, limiting its scalability in large-scale credit applications. The model's performance degrades notably in high-dimensional spaces, demonstrated by our drop from 72% to 61% with increased features. The algorithm's strong dependence on local data density and distance metrics makes it sensitive to noise and sparse regions in the feature space, which can affect its reliability.

7. Conclusion

Our study set out to determine whether machine learning models trained solely on readily available financial data could outperform traditional statistical models in predicting borrower default within a two-year span. Through comparative evaluation, we found that both k-NN (accuracy: 76.6%) and neural networks (accuracy: 75.9%) surpassed our logistic regression baseline (accuracy: 71.85%) in predictive accuracy, with k-NN providing the highest performance gain of 4.75%.

While logistic regression remains the most interpretable, k-NN provides moderate transparency through case-based decisions, making it a strong alternative. Neural networks, though accurate, lack interpretability for regulated financial settings.

We conclude that through our results, machine learning models, especially k-NN, can be trained on standard financial indicators more accurately, and in some cases, more transparently to assess borrowers' risk than traditional methods. The practical implication is clear: lending institutions can improve risk predictions without requiring complex or proprietary data. Future work should focus on scaling these models for real-time deployment and integrating fairness-aware algorithms to ensure ethical credit decisions.

References

- [1] E. Costa e Silva, I. C. Lopes, A. Correia, and S. Faria, "A logistic regression model for consumer default risk," *Journal of Applied Statistics*, vol. 47, no. 13–15, pp. 2879–2894, May 2020, doi: <https://doi.org/10.1080/02664763.2020.1759030>.
- [2] V. Moscato, A. Picariello, and G. Sperl , "A benchmark of machine learning approaches for credit score prediction," *Expert Systems with Applications*, vol. 165, p. 113986, Mar. 2021, doi: <https://doi.org/10.1016/j.eswa.2020.113986>.

-
- [3] S. Shi, R. Tse, W. Luo, S. D’Addona, and G. Pau, “Machine learning-driven credit risk: a systemic review,” *Neural Computing and Applications*, vol. 34, no. 17, pp. 14327–14339, Jul. 2022, doi: <https://doi.org/10.1007/s00521-022-07472-2>.
 - [4] A. A. Hussin Adam Khatir and M. Bee, “Machine Learning Models and Data-Balancing Techniques for Credit Scoring: What Is the Best Combination?,” *Risks*, vol. 10, no. 9, p. 169, Aug. 2022, doi: <https://doi.org/10.3390/risks10090169>.
 - [5] J. Liu, S. Zhang, and H. Fan, “A two-stage hybrid credit risk prediction model based on XGBoost and graph-based deep neural network,” *Expert Systems with Applications*, vol. 195, p. 116624, Jun. 2022, doi: <https://doi.org/10.1016/j.eswa.2022.116624>.