Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 9
AIDML 2025

Comparative Analysis of Traditional Statistical and Machine Learning Approaches in Credit Scoring Applications

Leung Hon Sum

HSBC School of Business, Peking University, Shenzhen, China

843718560@qq.com

Abstract. This paper compares the performance of traditional statistical approaches, such as logistic regression (LR), and machine learning approaches, like support vector machines (SVMs), in credit scoring. In this paper, a dataset is simulated containing borrower characteristics like income, wealth, repayment history, length of requested loans, and total debt, which can be altered to represent different macroeconomic scenarios. Mathematica is used to train or fit and ultimately test both LR and SVM models on the dataset, focusing on evaluation metrics such as accuracy, precision, and recall to assess their performance. Results show that SVM consistently outperforms LR, with recall being 31.8% higher, precision being 15.6% higher, and accuracy being only 4.6% higher. This suggests that banks should consider implementing machine learning methods for credit scoring, as long as they have access to large datasets and sufficient computational power. Traditional approaches like LR should not be dismissed, as they offer transparency and interpretability, which are essential for financial institutions due to the fact that they are regulated entities.

Keywords: Traditional Statistical; logistic regression; machine learning; support vector machines; credit scoring.

1. Introduction

Machine learning, being one of the most discussed topics during recent years, is widely agreed to be the third major technological revolution in modern economics. The potential of this technological advancement is uncertain, there are uncountable fields that either have direct or indirect relation to this technology; Machine learning has been gradually developing its link to our daily lives, with help of large data, from as minimal as providing personalized recommendation during online shopping, to as important as helping banks to decide whether an amount of a loan should be issued to an individual or a firm. In recent years, banks have been placing higher emphasis on using machine learning models, or more advance and technical case, artificial intelligence; to adapt to the traditional operations of banking system. Machine learning technologies can improve banks' ability to achieve four key outcomes: higher mark up, scaled personalization, distinctive omnichannel experiences and rapid innovation cycles [1]. Furthermore, banks have been increasing its attention to lowering credit risks, with the help of machine learning models. Without using big data and machine learning method, traditional statistical risk analysis models are far more cumbersome. According to Sadok et al., the employee of the risk department needs to first process the data, then fit them into designated models for testing and finding correlation between predictors, finally fitting the remaining variables into the score model [2]. With sufficient computational power and data, machine learning models can be trained to replace the necessity of human labour in this process, and perform a better result. In general, there are three types of machine learning models that can perform the technique for credit scoring, namely DT, SVM and MLP; however, SVM generates better results in terms of accuracy, recall and precision [3].

This paper aims to compare traditional statistical approaches, like logistic regression (LR), with machine learning techniques, specifically support vector machines (SVMs), in credit scoring applications. It uses adaptable datasets based on borrowers' financial characteristics, and the models are implemented in Mathematica; ultimately, their performance is compared using accuracy, precision, and recall metrics. Results show that SVMs consistently outperform logistic regression

under those three metrics, indicating that banks should place more emphasis on machine learning for credit scoring. However, adopting these advanced techniques requires large datasets and computational power, which is increasingly accessible due to technological advancements. Despite machine learning's advantages, traditional approaches shouldn't be redundant as they offer transparency and interpretability, which is crucial for regulated financial institutions.

2. Credit Scoring Definition

2.1. General Definition

According to the book 'Intelligent Credit Scoring' [4], it defines the event of default as being a prediction horizon, and its rate is depending on serial events, such as characteristic attributes, pools, rating grades or segments. The profit of banks largely dependent on the loans it make to borrowers, which defaulting from the loans increases the volatility of banks profit, and thus the risk of the bank; and thus under the three objective of banks, namely liquidity, security and profitability, banks need to offer certain degree of weighting to the objectives, then optimize.

2.2. Supervised Learning

Credit scoring is a supervised learning problem [5], it is a binary classification scenario, and it's objective is to classify good or bad borrowers; it helps banks minimises lost, through not lending to those classified bad - once observed the characteristic of the specific individual. However, it is not always the case that the lost can be minimised, which we need appropriate model, up-to-date data, a well defined and applicable decision boundary and etc.

The process of credit scoring requires a sequential step: (1) Collecting dataset. (2) Process the dataset (only required for traditional statistical approach). (3) Define the class label. (4) Train the machine learning model with the class label (it finds the optimised decision boundary) / Find the optimising line, hyper- plane or any sort of method that separates the classes for traditional statistical approach. (5) Test the model with testing set (a separate dataset), it is usually the dataset that the result is already shown, else we couldn't identify the confusion matrix (true positive, true negative and etc.), and thus we couldn't identify the evaluation metrics (accuracy, precision and recall). (6) Evaluate the result and thus decide which model to use, and what data can it be applied upon.

2.3. Importance of Credit Scoring

Credit scoring benefits both lenders and borrowers. However, the failure of credit scoring is more detrimental. If a loan is issued to a bad borrower, when default, not only bank faces a great loss on it's security, liquidity and profitability; but also it will be a severe burden for the borrower. A lose-lose situation that we want to avoid, thus we need a way to relate the economic and personal situation to accurately quantify the credit risk of the individual [6].

3. Statistical Model and Machine Learning Model in Credit Scoring

3.1. Statistical Model in Credit Scoring- Logistic Regression Model

For logistic regression, it is a model that takes in the value of independent variables x (categorical or continuous), in our case are those variables related to the characteristics of borrowers, and which its dependent variable y is normally a binary output (nominal or ordinal); the output of the dependent variable in this case is then 1 or 0, assigning a good creditor (non-default ones) the value 1, else 0 [7].

The equation of logistic regression in probability form is given by:

$$P(Y = 1|x) = \frac{1}{1 + exp(\alpha_0 + \alpha^T x)}$$

$$P(Y = 0|x) = 1 - P(Y = 1|x) = \frac{exp(\alpha_0 + \alpha^T x)}{1 + exp(\alpha_0 + \alpha^T x)}$$

In our case, $x \epsilon R^n$is the feature vector, and $P(Y = 1|x)$is the probability that classifies individual with feature x as a good borrower, otherwise a bad one. For {α0, α}, they are the parameters estimated by MLE using the simulated data (only the training set).

3.2. Machine Learning Model in Credit Scoring- Support Vector Machines (SVM)

Support vector machine is a model that maps a hyperplane (a decision boundary) to separate classes in a high dimensional feature space [5]. It is an optimizing algorithm with four basic concepts [8]: (1) The separating hyperplane. (2) The maximum hyperplane. (3) The soft margin. (4) The kernel function.

The optimal hyperplane algorithm [9] is found through:

The set of labeled training patterns:

$$(y_1, \mathbf{x}_1), ..., (y_n, \mathbf{x}_n) \qquad y_i \in \{-1,1\}$$

is said to be linearly separable if there exists a vector **w** and a scalar b such that the inequalities:

$$\mathbf{w}\mathbf{x}_i + b \geq 1 \quad \text{if } y_i = 1$$

$$\mathbf{w}\mathbf{x}_i + b \leq -1 \quad \text{if } y_i = -1$$

where the optimal hyperplane (w_o, b_o) exists in-between the classes, and is the arguments that maximises the distance:

$$\rho(w_0, b_0) = \frac{2}{|w_0|} = \frac{2}{\sqrt{w_0 \times w_0}}$$

thus the optimal hyperplane is the unique one that minimizes $\boldsymbol{w} \cdot \boldsymbol{w}$ under the constraint given.

3.3. Visualisation of SVM and LR

The first illustration (on the left) in Fig.1 shows the plot and decision boundary of the LR model, since the definition and application of LR is to find a best line to separate the classes through minimizing the log-loss, in the plot we can see the decision boundary (the red line) is a straight line separates the two classes (blue and green dots); whereas for SVM model (second illustration), it tries to find the best hyperplane that separate classes with the maximum margin, the margin is the distance between the decision boundary and the closest points from each class called support vectors (definition and application of SVM). For SVM, it is capable of finding non-linear decision boundaries using kernel functions (such as the Radial Basis Function will be used in this project), thus we can see the decision boundary is non-linear.

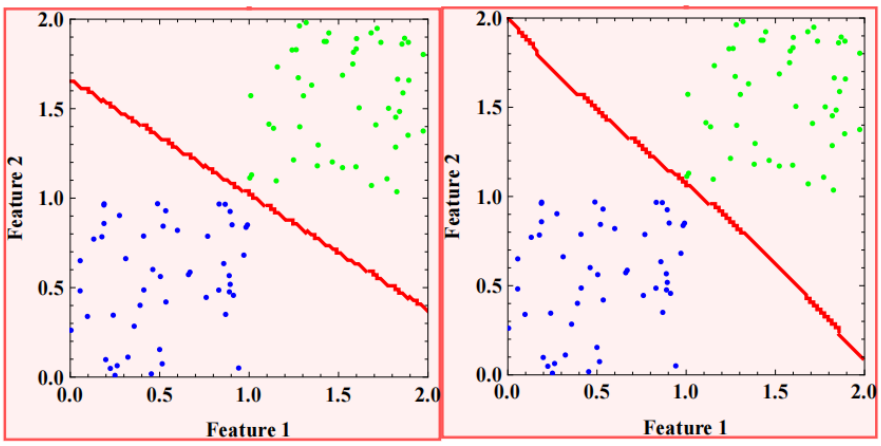

Fig 1. Visualising SVM and LR through decision boundary

4. Method

4.1. Simulating Characteristic Variable

Before comparing the model, we need to first create a dataset. Since our objective is to compare whether under the same dataset, which model (traditional or machine learning) performs better in terms of the evaluation statistics. Thus we simulate income, wealth, repayment history, length of requested loans and individual's total debt to include into the credit scoring evaluation. The dataset that we created can be easily altered under different circumstances, for example, in different state of a macroeconomic cycle; the alteration involves either pick a different distribution (e.g. normal distribution instead of log-normal ones) to represent the characteristic of interest, or changing the mean, standard deviation and any other scale or shape parameter of the distribution.

4.1.1. Simulating income distribution

Simulate the income distribution as a log-normal distribution. A log normal distribution is a positively skewed distribution, and is common to use to simulate the distribution of income [10]; an alternative to simulate income distribution is using the gamma distribu- tion. Since we will use gamma distribution later for wealth distribution, it can provide diversity in here for the project to use a log normal one, and thus the simulation becomes more interesting, and thus the result generate by different model then becomes interesting.

The lognormal distribution takes the form in probability density function [11]:

$$f(x) = \frac{1}{x(2\pi\sigma^2)} exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

Where μ is the mean of the natural logarithm of the variable, σ is the standard deviation of the natural logarithm of the variable, and x is the variable of interest (income in this case).

4.1.2. Simulating distribution of wealth and total debt

Simulate the wealth and total debt distribution as a gamma distribution, a gamma distribution is a 2-parameter frequency distribution given by the equation. Since the shape parameter k can be approximate to a fixed saving propensity [12], which is a great approxima- tion for factors like wealth, total debt, or any investment and etc.

The gamma distribution takes the form in probability density function [13]:

$$f(x) = \frac{1}{\left(\theta^k \Gamma(k)\right)} x^{k-1} e^{-\frac{x}{\theta}}$$

Where k is the shape parameter, θ is the scale parameter, $\Gamma(k)$ is the usual gamma function evaluated at k, and x is the variable of interest (wealth or total debt in this case). The distribution is positively skewed, and the skewness is inversely correlated with the shape factor k.

4.1.3. Simulating repayment history

Simulate the repayment history distribution as a beta distribution. We want to simulate repayment of debt given the parameters of the function into a probability. An alternative is to use binomial distribution (to simulate probability), however, binomial distribution is discrete, so it might not be applicable in our case.

The beta distribution takes the form in probability density function [14]:

$$f(x): prob(x|\alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\beta)}$$

Where B is the beta function:

$$B(\alpha,\beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1}dt$$

Where α and β are the shape parameters, B(α,β) is the beta function evaluated at α and β, and x is the variable of interest.

4.1.4. Simulating length of requested loans

Simulate the requested loan into a exponential distribution (decaying one). Length of the loan affects the ability to repay, and longer loans are riskier, and thus less people are willing to request to hold a longer period of loans.

The exponential distribution takes the form in probability density function:

$$f(x) = \lambda exp(-\lambda x)$$

where λ is the rate parameter, and x is the variable of interest (length of requested loans).

4.2. Combining the simulation in to a matrix

For each individual i, it's corresponding characteristic takes the form (table representation and matrix representation). The subscripts of 'a' is not only representing the row and column of the matrix entry; it takes the form of aij, where i is the index assigned to the

numbering of individual, and j corresponds to the characteristic according to their descending order in the table (say income is assigned as 1 and total debt is assigned as 5), all the data generated will then be imported into both SVM and LR model, trained using code included in programme of Mathematica.

Individual *i* / Characteristic	Income	Wealth	Repayment	Length	Total Debt
Ind 1	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}
...	...	...	...	...	...
Ind *N*	a_{N1}	a_{N2}	a_{N3}	a_{N4}	a_{N5}

$$\leftrightarrow \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ \dots & \dots & \dots & \dots & \dots \\ a_{N1} & a_{N2} & a_{N3} & a_{N4} & a_{N5} \end{pmatrix}$$

Using the random variate command in Mathematica, we could easily generate this N by 5 matrix, for this project, we will first generate 1000 (N=1000) individuals, and use to train or fit the model.

4.3. Assigning labels to individuals based on characteristics

Before inputting the characteristics into the model, we first need to define which individual can be considered to successfully repaying the loan; this is the process of assigning credit scores to individuals. We use some intuition here, and this intuition can be varied upon researcher's own objective indication that an individual is a good borrower. For this project, we have 4 restrictions to verify a good creditor, else a bad creditor.

Objective Function	Threshold	Value assigned
$\frac{\text{Income of individual } i}{\text{Wealth of individual } i * 1000}$ (ratio)	0.2	1 if above threshold, else 0
Repayment history (probability repaid)	0.5	1 if above threshold, else 0
Length of requested loans (time)	36	1 if below threshold, else 0
$\frac{\text{Total Debt of individual } i}{\text{Wealth of individual } i}$ (ratio)	0.6	1 if below threshold, else 0

For the first objective function, wealth is multiplied by 1000, because income is in unit of 1, but wealth and debt is in unit of 1000. Only if the individual satisfies all 4 restrictions (above or below the threshold), then it is valued that the individual will be a good creditor. It can also be calculated using $\prod_{i=1}^{4} V_i$, where V is the valuation under each restriction, and i is the index of the objective function (can be called as restriction); $\prod_{i=1}^{4} V_i = 1$ indicates good creditor.

After assigning labels to individuals, we combine the dataset of each individual and it's score of being good or bad creditor; we let the score take a new column entry following the last characteristic. Then we have this N by 6 matrix consisting of:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & \left(\prod_{i=1}^{4} V_{i1}\right) \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ a_{N1} & a_{N2} & a_{N3} & a_{N4} & a_{N5} & \left(\prod_{i=1}^{4} V_{iN}\right) \end{pmatrix}$$

4.4. Setting up Logistic Regression model and SVM model

4.4.1. Split the dataset into training and testing sets

From the matrix (above) that we created for 1000 individuals and its corresponding valuation, we will only take some portion of the dataset, say 80%, to be the training set; the remaining 20% will then be the testing set, to test the fit of the model, and offers evaluation to the model which will be discussed in the result section. The ratio of training set can vary, but most of the time it is better to train or fit the model with larger number of dataset, however, it also requires that the number of testing set is sufficient to distinguish the performance of the model.

4.4.2. Set up the model (SVM and LR) using training set

In Mathematica, we could use the codes embedded to fit or train or model. To perform fitting LR (using dataset) in Mathematica, the steps are the following:(1) Input the matrix defined from the above sections to the model. (2) Define the array of variable 'x' with the same length as the number of input features. (3) Call 'LogitModelFit', which means that Mathematica fits a LR model to the training data and returns to the model.

To perform training SVMs (using dataset) in Mathematica, the steps are the following:(1) Input the matrix defined from the above sections to the model. (2) Define the SVM model, for this project, it defines the SVM classifier with the RBF kernel type. (3) Train the SVM model using the 'classify' function.

The RBF (radial basis function) kernel is selected, because it is a popular choice for SVM classification, and it is capable of modeling complex, non-linear decision boundaries between the

classes; when comparing which kernel type to choose (RBF or polynomial), RBF is suitable when dataset is large enough [8]. The selection of kernel will have effect on accuracy, precision and recall of classification; we could let Mathematica to automatically choose kernel type to improve the three evaluation metrics of the model as well, which then requires a greater computational power to do so, if the technology used allows.

5. Results

5.1. Checking the balance and imbalance of dataset

The definition of a balanced dataset is one in which the class distribution is roughly equal (class 0 and 1), meaning that there are approximately the same number of instances for each class, else imbalanced.

$$\text{Class proportions} = \frac{\text{Frequency in class 0}}{\text{Frequency in class 0} + \text{Frequency in class 1}} = \frac{\text{Frequency in class 0}}{\text{Total frequency}}$$

```
Class Frequencies: {{0, 899}, {1, 101}}
Class Proportions: {0.899, 0.101}
The dataset is imbalanced.
```

From the illustration above, we can see that for the random 1000 individuals (total frequency) that we generated, there is 89.9% classified as bad borrower (assigned class 0); the range that defines a balance dataset in this project is if the class proportion is in between 0.4 and 0.6, where 0.899 is above the upper bound of the range, thus the dataset is imbalanced.

5.2. Comparison of the performance of SVM and LR

5.2.1. Performance Metrics definition and result

TP, FP, TN and FN (elements in a confusion matrix) will be used to distinguish the accuracy (proportion correctly classified - PCC), precision and recall. Where TP, FP, TN, and FN are true positives, false positives, true negatives, and false negatives:

True positive (TP): when the classifier correctly predicts a positive outcome (class 1) when the true outcome is positive (class 1).

False positive (FP): when the classifier incorrectly predicts a positive outcome (class 1) when the true outcome is negative (class 0).

True negative (TN): when the classifier correctly predicts a negative outcome (class 0) when the true outcome is negative (class 0).

False negative (FN): when the classifier incorrectly predicts a negative outcome (class 0) when the true outcome is positive (class 1).

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

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

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

5.2.2. Performance Metrics of LR (3 results)

We run the full code (from generating dataset to evaluating the model) 3 times and get this result:

```
Accuracy: 0.915      Accuracy: 0.915     Accuracy: 0.93
Recall: 0.5625       Recall: 0.541667    Recall: 0.68
Precision: 0.857143  Precision: 0.684211 Precision: 0.73913
```

Take the geometric mean of these 3 results for performance of LR:

$$\text{Accuracy} = \sqrt[3]{0.915 \times 0.915 \times 0.93} = 0.91997 \text{ (5 d.p.)}$$
$$\text{Recall} = \sqrt[3]{0.5625 \times 0.54167 \times 0.68} = 0.59173 \text{ (5 d.p.)}$$
$$\text{Precision} = \sqrt[3]{0.85714 \times 0.68421 \times 0.73913} = 0.75681 \text{ (5 d.p.)}$$

5.2.3. Performance Metrics of SVM (3 results)

We run the full code (from generating dataset to evaluating the model) 3 times and get this result:

```
Accuracy: 0.95       Accuracy: 0.97   Accuracy: 0.97
Recall: 0.769231     Recall: 0.818182 Recall: 0.862069
Precision: 0.833333  Precision: 0.9   Precision: 0.925926
```

Take the geometric mean of these 3 results for performance of SVM:

$$\text{Accuracy} = \sqrt[3]{0.95 \times 0.97 \times 0.97} = 0.96329 \text{ (5 d.p.)}$$
$$\text{Recall} = \sqrt[3]{0.76923 \times 0.81818 \times 0.86207} = 0.81561 \text{ (5 d.p.)}$$
$$\text{Precision} = \sqrt[3]{0.83333 \times 0.9 \times 0.92593} = 0.88555 \text{ (5 d.p.)}$$

5.3. Compare the difference in performance

In this section, since the performance metrics of SVM are always greater than LR, we use the number of 3 different metrics of SVM minus the corresponding number of LR, then dividing it by the mean value of the number to get a percentage difference.

$$\text{Accuracy} = \frac{(\text{Accuracy of SVM} - \text{Accuracy of LR})}{\left(\frac{\text{Accuracy of SVM} + \text{Accuracy of LR}}{2}\right)} = \frac{0.96329 - 0.91997}{\left(\frac{0.96329 + 0.91997}{2}\right)} = 0.04601 \text{ (5 d.p.)} = 4.6\% \text{ (1 d.p.)}$$

$$\text{Recall} = \frac{(\text{Recall of SVM} - \text{Recall of LR})}{\left(\frac{\text{Recall of SVM} + \text{Recall of LR}}{2}\right)} = \frac{0.81561 - 0.59173}{\left(\frac{0.81561 + 0.59173}{2}\right)} = 0.31816 \text{ (5 d.p.)} = 31.8\% \text{ (1 d.p.)}$$

$$\text{Precision} = \frac{(\text{Precision of SVM} - \text{Precision of LR})}{\left(\frac{\text{Precision of SVM} + \text{Precision of LR}}{2}\right)} = \frac{0.88555 - 0.75681}{\left(\frac{0.88555 + 0.75681}{2}\right)} = 0.15677 \text{ (5 d.p.)} = 15.6\% \text{ (1 d.p.)}$$

6. Discussion

The main purpose of this paper is to compare the performance in traditional statistical approaches and machine learning methods, thus it is only interesting in the final 3 metrics that we calculated from above. The percentage differences (of SVM and LR) for recall is 31.8% (highest in the metrics),

suggesting that SVM outperforms LR in recall; whereas for precision, it is 15.6% (second highest) better than LR. The accuracy of two models are similar, which is only 4.6% (lowest) difference.

Putting aside numerical part, it is more important to consider which metrics of the models affect banks' three objectives (liquidity, security and profitability) the most. For credit risk scoring, precision and recall are often considered more important than accuracy, due to the potential costs associated with mis-classifying creditworthy (positive) and non-creditworthy (negative) customers. In addition, since the training and testing sets come from 5 same distribution, the accuracy will always be high; it is consistent. If for instance, the testing set is from a same type of distribution but different parameter, then it is hard for the model to generalize, which leads to poor performance (on accuracy) - it is always the case in the reality that the predictive power of model is low due to the fact of unpredictable economics circumstance (shocks).

Back to recall and precision, precision is the measure of proportion of true positives (predicted non-default) among the case predicted as positive. A higher precision indicates that the classifier is good at identifying creditworthy customers, which in turn increase the banks' profit, since banks can then offer more loans (to the good borrower) according to the 'guide' of the model. For recall, it measures the proportion of true positives among the actual positive. Higher recall indicates that the classifier is good at identifying most creditworthy customers and minimizing false negatives. In credit risk scoring, false negatives could lead to missed business opportunities by denying loans to the actual good borrowers (false negatives).

After comparing the evaluation metrics for traditional statistical approach (LR) and machine learning approach (SVM), the latter outperforms the former. Banks should put more emphasis into applying their data to use the machine learning method. However, it requires prior that the dataset is large enough to train the model. Furthermore, even machine learning can easily deal with large dataset, but it also requires sufficient computational power; it will not be a problem, since we know technology is consistently developing (e.g. quantum computer or supercomputers), and in the future, much complex model can be developed - in this project our goal is also to show that technological advancement (machine learning) generates economics benefits, for banks and ultimately the financial system. On the other hand, we should not avoid using traditional approach, due to the fact that the process of fitting the model is more transparent and interpretable [5]. For financial institutes (banks), they are regulated entities, which then needs the transparency and interpretability in their decision and models. Ultimately, we should employ the benefits of traditional approach, and also keep up with the disruption of technology, and thus optimize.

7. Summary

While this research indicates the superiority of SVM over LR in credit scoring, there are some limitations to consider, such as the inherent advantage of SVM in handling complex and imbalanced data and the potentially unrealistic distributions used in the simulation. Future research could address these limitations by comparing traditional statistical approaches and machine learning methods for the same model, adapting the simulation to fit specific regions or macroeconomic conditions, and incorporating shocks to the data for a more realistic comparison. To conclude, this study demonstrates the potential of machine learning methods like SVM to improve credit scoring performance, provided that banks have access to the necessary resources, and highlights the importance of remaining open to incorporating machine learning techniques into credit risk assessment processes to optimize decision-making and ultimately strengthen the financial system.

References

[1] Biswas, S., Carson, B., Chung, V., Singh, S. and Thomas, R., 2020. AI-bank of the future: Can banks meet the AI challenge. New York: McKinsey & Company.

[2] Sadok, H., Sakka, F., & El Maknouzi, M. E. H., 2022. Artificial intelligence and bank credit analysis: A review. Cogent Economics & Finance, 10(1), 2023262.

[3] Ghodselahi, A. and Amirmadhi, A., 2011. Application of artificial intelligence techniques for credit risk evaluation. International Journal of Modeling and Optimization, 1(3), p.243.

[4] Siddiqi, N., 2017. Intelligent credit scoring: Building and implementing better credit risk scorecards. John Wiley & Sons.

[5] Dastile, X., Celik, T. and Potsane, M., 2020. Statistical and machine learning models in credit scoring: A systematic literature survey. Applied Soft Computing, 91, p.106263.

[6] Avery, R. B., Calem, P. S., & Canner, G. B., 2004. Consumer credit scoring: do situational circumstances matter?. Journal of Banking & Finance, 28(4), 835-856.

[7] Gouvêa, M.A. and Gonçalves, E.B., 2007, May. Credit risk analysis applying logistic regression, neural networks and genetic algorithms models. In POMS 18th annual conference.

[8] Prajapati, G.L. and Patle, A., 2010, November. On performing classification using SVM with radial basis and polynomial kernel functions. In 2010 3rd International Conference on Emerging Trends in Engineering and Technology (pp. 512-515). IEEE.

[9] Cortes, C., & Vapnik, V., 1995. Support-vector networks. Machine learning, 20, 273-297.

[10] Salem, A.B. and Mount, T.D., 1974. A convenient descriptive model of income distribution: the gamma density. Econometrica: journal of the Econometric Society, pp.1115-1127.

[11] Crow, E.L. and Shimizu, K., 1987. Lognormal distributions. New York: Marcel Dekker.

[12] Chakraborti, A. and Patriarca, M., 2008. Gamma-distribution and wealth inequality. Pramana, 71, pp.233-243.

[13] Thom, H.C., 1958. A note on the gamma distribution. Monthly weather review, 86(4), pp.117-122.

[14] Johnson, N.L., Kotz, S. and Balakrishnan, N., 1994. Beta distributions. Continuous univariate distributions. 2nd ed. New York, NY: John Wiley and Sons, pp.221-235.