LIV Simpósio Brasileiro de Pesquisa Operacional
Juiz de Fora, MG - 8 a 11 de novembro de 2022

MODELAGEM DE UM CLASSIFICADOR SVM PARA ANÁLISE DE RISCO DE CRÉDITO

André Aoun Montevechi
andremontevechi@gmail.com.br
Rafael de Carvalho Miranda
rafael.miranda@unifei.edu.br
André Luiz Medeiros
andremedeiros@unifei.edu.br
João Victor Soares do Amaral
joao.amaral@unifei.edu.br
José Arnaldo Barra Montevechi
montevechi@unifei.edu.br

Universidade Federal de Itajubá (UNIFEI)
Avenida BPS, 1303, Itajubá – MG

RESUMO

A concessão de crédito é uma atividade vital da indústria financeira. Para o funcionamento e sucesso das instituições financeiras, assim como a manutenção do equilíbrio do sistema creditício, a modelagem de risco de crédito tem o papel de avaliar a probabilidade de inadimplência de potenciais devedores com base em dados históricos. Modelos e algoritmos de classificação apresentam uma abordagem interessante para esta finalidade. Neste contexto, este trabalho explora a modelagem e avaliação de um classificador para aferir o risco de crédito por meio de *Support Vector Machines* (SVM), um algoritmo supervisionado de *Machine Learning*, com auxílio de algoritmos de otimização de hiper-parâmetros e *resampling*. O classificador SVM proposto obteve excelente poder de discriminação, com métricas de performance superiores à abordagem padrão da regressão logística.

PALAVRAS CHAVE. Modelagem de Risco de Crédito. *Support Vector Machine (SVM)*. *Machine Learning*.

Tópicos: Gestão Financeira, Inteligência Computacional.

ABSTRACT

Granting credit is the most vital activity of the financial industry. For its success, as well as the balance of the credit system as a whole, credit risk modeling must be able to properly assess a potential client's probability of default based on historical data. Classification models and algorithms provide an interesting way to approach this problem. This paper implements and evaluates the performance of a supervised Machine Learning model, in this case a Support Vector Machine (SVM) classifier, in doing so with the aid of hyperparameter optimization and resampling algorithms. The proposed classifier obtained excellent discriminatory power, with higher performance metrics than the traditional logistic regression approach.

KEYWORDS. Credit Risk Modeling. Support Vector Machine (SVM). Machine Learning.

Paper topics: Financial Management, Computational Intelligence.

1. Introdução

A concessão de crédito é indissociável da indústria financeira [Coenen et al. 2020]. Diariamente, instituições financeiras liberam créditos para pessoas físicas ou jurídicas por diferentes meios, consolidando uma relação contratual obrigacional entre ambas as partes que assumem, respectivamente, os papéis de credor e devedor (ou tomador). Contudo, este procedimento traz um risco inerente para o credor: o não pagamento devido do valor emprestado.

Assim, credores enfrentam o desafio de avaliar o risco de inadimplência de possíveis tomadores. O desenvolvimento de soluções para tal é de suma importância, sendo crucial para o sucesso dos credores e para a manutenção do equilíbrio do sistema creditício. Um dos nomes que se dá a este processo é *Credit Scoring* (CS). Para isso, foram desenvolvidos diferentes modelos matemáticos que buscam determinar, com base na análise de dados históricos, se um empréstimo será pago ou não (o que se chama de classificação), e até mesmo estimar a probabilidade de inadimplência, conhecida como *Probability of Default* (PD).

Modelos clássicos de CS abordam este problema com o emprego de métodos estatísticos [Marqués et al. 2013], em especial a regressão logística (*logistic regression* ou LR), que gradualmente se tornou o padrão na indústria [Dumitrescu 2021]. Contudo, recentemente observou-se numerosos novos estudos da aplicação de variados métodos de *Machine Learning* (ML) com este propósito [Wang et al. 2020], trazendo novas formas de implementação destes modelos estatísticos, bem como outros algoritmos de maior sofisticação e complexidade. A evolução de ML tem tido grande impacto sobre as metodologias habituais de CS e cálculo da métrica PD, considerando que possibilita o uso de grandes quantidades de dados históricos para a modelagem [Lessmann et al. 2015] e tem gerado bons resultados.

Melhorias em frações percentuais para um modelo podem implicar em significativas economias [Baesens et al. 2003]. À vista disso, este artigo tem como objetivo detalhar o processo de modelagem de risco de crédito por meio da aplicação de um algoritmo supervisionado de ML para classificação chamado de *Support Vector Machine* (SVM), e avaliá-lo frente as métricas de performance adequadas e a metodologia padrão da LR. Partindo de uma base de dados históricos de concessão de crédito contendo informações a respeito de tomadores e seus respectivos contratos, propõe-se a programação de um classificador e a avaliação de sua performance mediante as métricas apropriadas.

Para cumprir com seus objetivos este trabalho está dividido em 6 seções. A seção 2 apresenta o referencial teórico. A seção 3 descreve brevemente os passos para a condução da presente pesquisa. A seção 4 apresenta a aplicação do método. A seção 5 discute os resultados, e, por fim, a seção 6 apresenta as conclusões.

2. Referencial teórico

2.1. SVM

Proposto por Vapnik [1999], o SVM é um dos métodos de ML mais populares dos últimos anos [Cao et al. 2020]. Desenvolvido inicialmente para problemas de classificação, o método busca o limite que separa os dados de treinamento em suas respectivas classes com o mínimo erro possível. Para isso, é definido o hiperplano com base em vetores construídos pelas observações que estão localizadas nos limites e os dados que se encontram em lados distintos do hiperplano são alocados em classes diferentes [Østergård et al. 2018].

Conforme Jeng [2006], o SVM tenta encontrar a margem máxima entre as duas classes pela combinação linear dos vetores, solucionando o problema por meio de uma programação quadrática com restrições lineares. Com a introdução do parâmetro $\mathcal{E}$, que representa a função de perda, este algoritmo também pode ser estendido à solução de problemas não lineares. A formulação matemática para o SVM, conforme Li et al. [2010], encontra-se disposta na Eq. 1:

$$\hat{y} = \hat{f}(X) = w\phi(X) + b \quad (1)$$

Sendo $X \in \Re^k$ o input k-dimensional, ϕ a transformação não-linear de $\Re^k$ para $\Re^h$, com $h > k$, e $\hat{f}(X)$ o hiperplano em $\Re^h$. Já w e b são coeficientes obtidos pela minimização da função de risco descrita pela Eq. 2.

$$R_{svm}(C) = C\sum_{i=1}^{n} C\left(\hat{f}(X_i), y_i\right) + \frac{1}{2}\|w\|^2 \quad (2)$$

No qual:

$$C\left(\hat{f}(X_i), y_i\right) = \begin{cases} \left|\hat{f}(X) - y\right| - \varepsilon & se\ \left|\hat{f}(X) - y\right| \geq \varepsilon \\ 0 & Caso\ Contrário \end{cases} \quad (3)$$

Solucionado o problema de minimização, a função de previsão final do método SVM é obtida pela Eq. 4 [Vapnik 1999].

$$f(X, a_i, a_i^*) = \sum_{i=1}^{n} (a_i - a_i^*) K(X, X_i) + b \quad (4)$$

Sendo X o novo input do modelo, $K(X, X_i)$ a função que representa a distância entre X e X_i, conhecida como Kernel, e a_i, a_i^* e b os coeficientes obtidos pela minimização da função de risco [Li et al. 2010]. Os principais componentes do SVM podem ser observados na Figura 1.

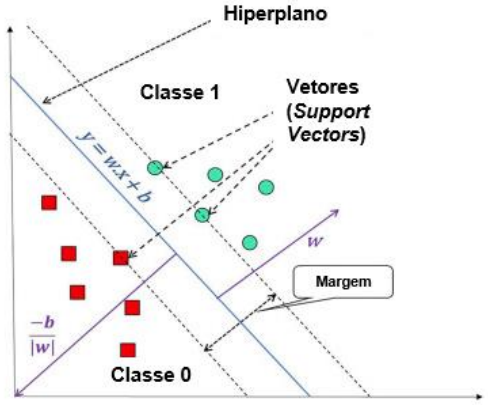


Figura 1 – Componentes do SVM
Fonte: Adaptado de Rani et al. [2022]

2.2 *Resampling*

Uma grande adversidade em problemas de classificação é o desbalanceamento de classes, quando o número de amostras de uma classe difere significativamente da outra. O desbalanceamento pode afetar negativamente a performance da classificação [Chen 2021], na medida em que se aumenta o risco da previsão preferir a classe majoritária [Onel 2018] no fenômeno conhecido como *overfitting*.

Marqués et al. [2013] examinaram o efeito de metodologias de *resampling* na performance de classificadores, encontrando melhoras no resultado em todas as instâncias testadas em relação à base desbalanceada. É necessário, portanto, antes do treinamento do algoritmo de classificação, lidar com o desbalanceamento de classes. Isto pode ser atingido de duas maneiras distintas no processo chamado de *resampling*: eliminando amostras pertencentes à classe majoritária (*undersampling*) ou adicionando amostras à classe minoritária (*oversampling*). Em ambos os casos, isto pode ser feito aleatoriamente (*random resampling*) ou com emprego de algoritmos que o façam de forma mais elaborada, já que com o *random undersampling* se arrisca

eliminar atributos importantes para a classificação e, com *random oversampling*, não se altera a variedade das amostras.

Para o experimento descrito neste artigo, consequentemente, foram abordados dois algoritmos de *resampling* na tentativa de lidar com o desbalanceamento de classes entre os dados: *Synthethic Minority Oversampling Technique* (SMOTE) e *Tomek links*. Respectivamente, algoritmos de *oversampling* e *undersampling*.

2.2.1. SMOTE

Este algoritmo de *oversampling* seleciona amostras de cada classe e seus vizinhos mais próximos (*nearest neighors*) para então criar novas amostras sintéticas – e não cópias – e incrementar a classe minoritária por meio da combinação dos atributos de suas amostras selecionadas e os pontos vizinhos [Ghorbani e Ghousi 2020].

2.2.2. *Tomek links*

Um *Tomek link* refere-se à ocorrência de amostras de diferentes classes que, contudo, são vizinhas mais próximas. Esta metodologia, quando empregada como forma de *undersampling*, identifica as instâncias em que isto ocorre e elimina a amostra pertencente à classe majoritária [Bach et al. 2019].

2.3. Otimização de hiper-parâmetros

Algoritmos de ML comumente enfrentam problemas de otimização [Wu et al. 2019]. Nestes problemas, a função de otimização é composta por múltiplos hiper-parâmetros definidos antes do treinamento do modelo e que afetam diretamente a forma com que ele se ajusta aos dados. Neste contexto, Oliveira et al. [2010] afirmam que a boa eficiência do SVM depende da escolha correta dos hiper-parâmetros. Para uma escolha apropriada, recorre-se ao processo de otimização de hiper-parâmetros (OHP).

Essencialmente, existem dois métodos de OHP: *manual search*, no qual se testam combinações de hiper-parâmetros manualmente; e *automatic search*, que conta com uso de algoritmos. Este estudo emprega o segundo método, mais particularmente um algoritmo de OHP chamado de *grid search*.

O *grid search* trata de um dos algoritmos de *automatic search* de mais fácil entendimento e implementação na etapa de OHP. O seu princípio é treinar o modelo de ML com possíveis combinações de hiper-parâmetros para chegar na melhor performance de acordo com uma métrica pré-definida [Wu et al. 2019]. Essencialmente, assim que são avaliadas cada uma das combinações, escolhe-se aquela com o melhor resultado para o modelo.

2.4. Métricas de avaliação

Para avaliar a performance de um modelo objetivamente, devem ser usadas métricas apropriadas para aferi-los [Chen 2021]. É preciso também entendê-las e interpretá-las para que não se corra o risco da validação de um modelo ineficaz.

2.4.1. AUC

Em problemas de classificação dicotômica, como é o caso do modelo de CS proposto (um classificador), as classes possuem duas categorias: positiva e negativa. Quando o modelo de classificação $h_i(x)$ prevê a classe de uma nova amostra, podem ocorrer quatro diferentes circunstâncias. Pode se concluir que uma amostra positiva é positiva (*true* positive ou TP) ou que ela é negativa (false negative ou FN) e, similarmente, que uma amostra negativa é positiva (false positive ou FP) ou negativa (*true* negative ou TN). Estes quatro casos compões a chamada matriz de classificação (Tabela 1).

Tabela 1 – Matriz de classificação

	Positivo (Y = 1)	Negativo (Y = 0)
Classificado como positivo ($h_i = 1$)	TP	FP
Classificado como negativo ($h_i = 0$)	FN	TN

Fonte: Adaptado de Shen et al. [2020]

Em classificadores tradicionais para CS, segundo Shen et al. [2020], AUC (*Area Under the Receiver Operating Characteristic Curve*) é uma forma efetiva para a avaliação de performance que, na prática, tem sido muito utilizada com esta finalidade. Corresponde à área abaixo da curva ROC (*Receiver Operating Characteristic Curve*) (Figura 2) que, por sua vez, é baseada na taxa de falsos positivos (*false positive rate* ou FPR), definida na Eq. 5, plotada no eixo das abcissas e a taxa de verdadeiros positivos (*true positive rate* ou TPR), definida pela Eq. 6, no eixo das ordenadas.

$$FPR = \frac{FP}{FP + TN} \tag{5}$$

$$TPR = \frac{TP}{TP + FN} \tag{6}$$

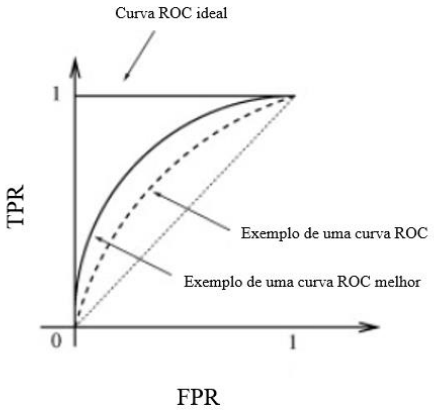


Figura 2 – Exemplo de curvas ROC

Fonte: Adaptado de Meyer-Baese e Schmid [2014]

A curva ROC ilustra graficamente o desempenho de um modelo de classificação binário na discriminação entre classes [Trucco et al. 2019]. Quanto mais ela se aproxima da curva ROC ideal, melhor é o desempenho do modelo. Portanto, quanto maior a área abaixo da curva (AUC), melhor é a performance do classificador [Shen et al. 2020].

Tabela 2 – Interpretações do valor de AUC

Valor de AUC	Interpretação
AUC = 0.5	Nenhuma discriminação (aleatório, e.g. jogar na moeda)
0.5 < AUC ≤ 0.6	Discriminação ruim
0.6 < AUC ≤ 0.7	Discriminação aceitável
0.7 < AUC ≤ 0.9	Discriminação excelente
AUC > 0.9	Discriminação exemplar

Fonte: Adaptado de Yang e Berdine [2017]

2.4.2. Precisão

A precisão de um modelo equivale ao percentual correto de previsões das amostras de uma classe. Pode ser calculada pelas Eqs. 7 e 8.

$$\text{Precisão (classe positiva)} = \frac{TP}{TP + FP} \tag{7}$$

$$\text{Precisão (classe negativa)} = \frac{TN}{TN + FN} \tag{8}$$

Para melhor ilustrar isto, como exemplo, quando um modelo preciso prevê que a classe de uma amostra será positiva, existe uma grande chance de a previsão estar correta, enquanto um modelo não preciso pode prever erroneamente muitas amostras como positivas.

2.4.3. *Recall*

Recall corresponde à capacidade de previsões corretas de amostras de uma classe pelo modelo de classificação, sendo calculado pelas Eqs. 9 e 10.

$$Recall\ (classe\ positiva) = \frac{TP}{TP + FN} \tag{9}$$

$$Recall\ (classe\ positiva) = \frac{FP}{FP + TN} \tag{10}$$

É possível notar que se tratam de métricas equivalente ao *true positive rate* (TPR) e *false positive rate* (FPR) do modelo e que são empregados na elaboração da curva ROC.

2.4.4. F1-*score*

A métrica do F1-*score*, por sua vez, avalia modelos de classificação ao combinar as métricas de precisão e *recall*, tendo sido desenvolvida para aferir bem casos com desbalanceamento de dados. Pode ser calculado conforme a Eq. 11.

$$\text{F1} - score = \frac{2 * \text{Precisão} * Recall}{\text{Precisão} + Recall} \tag{11}$$

3. Método de pesquisa

Para a elaboração do modelo classificador de CS proposto foi seguido o *framework* da Figura 3.

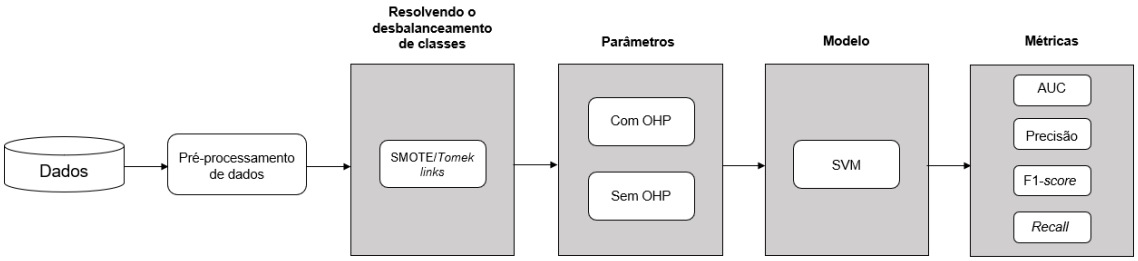


Figura 3 – *Framework* do processo

Primeiramente foi conduzida uma análise exploratória dos dados para se obter uma visão geral da base de dados e um melhor entendimento dos atributos das amostras e para que, na etapa de pré-processamento, fossem feitas as alterações necessárias em preparação para, em seguida, os processos de *resampling*, OHP e elaboração do modelo.

Na sequência serão descritas as etapas que compõem este *framework* em sua aplicação a base de dados utilizada neste trabalho para elaboração do modelo de classificação.

4. Aplicação

4.1. Descrição dos dados e pré-processamento

Foi utilizada uma base de dados de empréstimos *peer-to-peer* provida pela empresa estadunidense *LendingClub* e disponibilizada pela *Kaggle*, uma comunidade *online* para cientistas de dados e profissionais de ML.

A base de dados contém 20.000 amostras caracterizando diferentes empréstimos realizados pela empresa por meio de 13 variáveis, sendo as 12 primeiras os atributos dos tomadores e, a última, a classe, nossa variável dependente, discreta e binária, chamada de *bad_loan*, e que recebe o valor de 1 caso o empréstimo não tenha sido pago.

Como discutido, no processo de modelagem do risco de crédito, objetiva-se prever, embasados por dados históricos, se um novo tomador pagará ou não o empréstimo desejado. Sendo isto caracterizado pela variável *bad_loan*, que é discreta e binária, assumindo valores de 0 a 1, fica evidente que a base em questão apresenta um problema de classificação.

4.2. Pré-processamento

Analisando melhor os dados, pode-se também atestar se a base se encontra balanceada ou não. Das 20.000 amostras disponíveis, evidenciou-se que 16.000 caracterizam empréstimos devidamente pagos (classe 0) e os 4.000 restantes caracterizam empréstimos não devidamente pagos (classe 1).

Também foram eliminadas as amostras com a falta de atributos. Os resultados foram 19.846 amostras restantes, sendo 15.875 pertencentes à classe 0, 3.971 pertencentes à classe 1, mantida a proporção entre as classes. É certo, portanto, sendo a classe majoritária quatro vezes maior do que a minoritária, que a base se encontra desbalanceada e a modelagem pode ser prejudicada no caso da ausência de medidas corretivas.

4.3. Resolução do desbalanceamento de classes

Ante o desbalanceamento de classes entre os dados utilizados, evidente na Figura 4 (a), foi empregada uma combinação dos algoritmos SMOTE e *Tomek links*, implementado em *Python*, para efetuar o balanceamento das classes.

O resultado foi uma nova base, ilustrada pela Figura 4 (b), agora balanceada, com 29.556 amostras, divididas igualmente entre as classes.

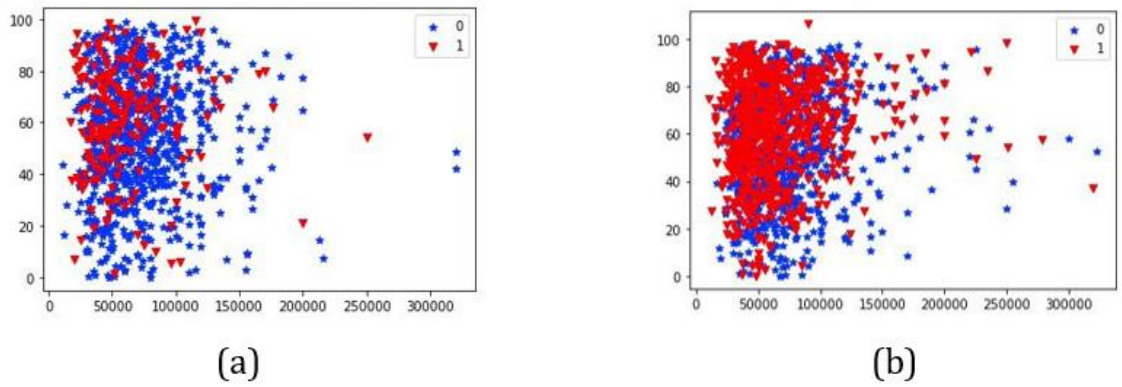


Figura 4 – Distribuição para 2 atributos da base de dados (a) desbalanceada e (b) balanceada por meio de algoritmos de *resampling* (*sampling rate: 0.15)*

4.4. Encontrando os hiper-parâmetros

Para a escolha dos hiper-parâmetros, foi implementado um algoritmo para *grid search* em *Python* na etapa de OHP. Primeiramente, o Kernel escolhido foi *Radial-Basis Function* (RBF), definido por Niklis et al. [2014] na Eq. 12.

$$\text{RBF}: K(x_i, x_j) = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right) \tag{12}$$

Foram então feitas combinações para diferentes valores dos hiper-parâmetros C (Eq. 3) e γ (Eq. 12), formando a *grid* da Figura 5.

(C, γ)

(0.1, 0.001)	(0.1, 0.01)	(0.1, 0.1)	(0.1, 1)
(1, 0.001)	(1, 0.01)	(1, 0.1)	(1, 1)
(10, 0.001)	(10, 0.01)	(10, 0.1)	(10, 1)
(100, 0.001)	(100, 0.01)	(100, 0.1)	(100, 1)

Figura 5 – Esquematização das combinações testadas pela *grid search*

Os valores para os hiper-parâmetros C e γ que formaram a melhor combinação conforme o algoritmo são, respectivamente 100 e 0.1.

4.5. Resultados

Assim como na etapa de *resampling* e na OHP, o algoritmo de SVM também foi implementado em *Python*. Foram construídos dois modelos, com e sem otimização de hiper-parâmetros. Também foi construído um modelo de LR para a melhor validação dos resultados obtidos, sabendo-se que essa metodologia constitui o *benchmark* para CS na indústria financeira.

Para todos os modelos, dois terços da bases de dados balanceada foi destinada ao treinamento e o restante para os testes e avaliação de performance.

5. Resultados

Analisando a Tabela 3, é possível evidenciar a importância de boas escolhas para os hiper-parâmetros do algoritmo do SVM, uma vez que, implementando-o sem a etapa de OHP, com os hiper-parâmetros padrão do algoritmo, verificaram-se valores reduzidos para todas as métricas de performance propostas para a avaliação do modelo.

Tabela 3 – Métricas de precisão, *recall* e F1-*score* para o modelo SVM sem OHP

Classe	Precisão	*Recall*	F1-*score*
0	0.58	0.56	0.57
1	0.57	0.58	0.57

Não surpreendentemente, observando a Figura 6 (a), atesta-se um baixo valor de AUC para este modelo, valor que, pouco superior a 0.5 – o que segundo a Tabela 2 relata poder discriminatório ruim – indica que o modelo não é muito superior a escolhas randômicas.

Já para o modelo de LR, evidencia-se uma melhoria na performance conforme as métricas escolhidas (Tabela 4), o que é confirmado pelo aumento para AUC no caso deste modelo, conforme a Figura 6 (b) que, atestando seu maior poder de classificação – descrito como aceitável pela Tabela 2 – demonstra a utilidade da aplicação de algoritmos de LR em ML para a modelagem do risco de crédito e parte do porquê esta metodologia tornou-se padrão na indústria financeira.

Tabela 4 – Métricas de precisão, *recall* e F1-*score* para o modelo de LR

	Precisão	*Recall*	F1-*score*
0	0.67	0.68	0.68
1	0.68	0.68	0.68

Por fim, com a etapa de OHP antes do treinamento do modelo SVM, observaram-se resultados promissores, havendo um aumento significativo nos valores das métricas de avaliação, conforme a Tabela 5.

Tabela 5 – Métricas de precisão, *recall* e F1-*score* para o modelo SVM com OHP

	Precisão	*Recall*	F1-*score*
0	0.81	0.79	0.80
1	0.79	0.81	0.80

Tal melhoria é melhor ilustrada pela Figura 6 (c), que traz para este modelo um valor de AUC que, segundo a Tabela 2, apresenta poder discriminatório excelente.

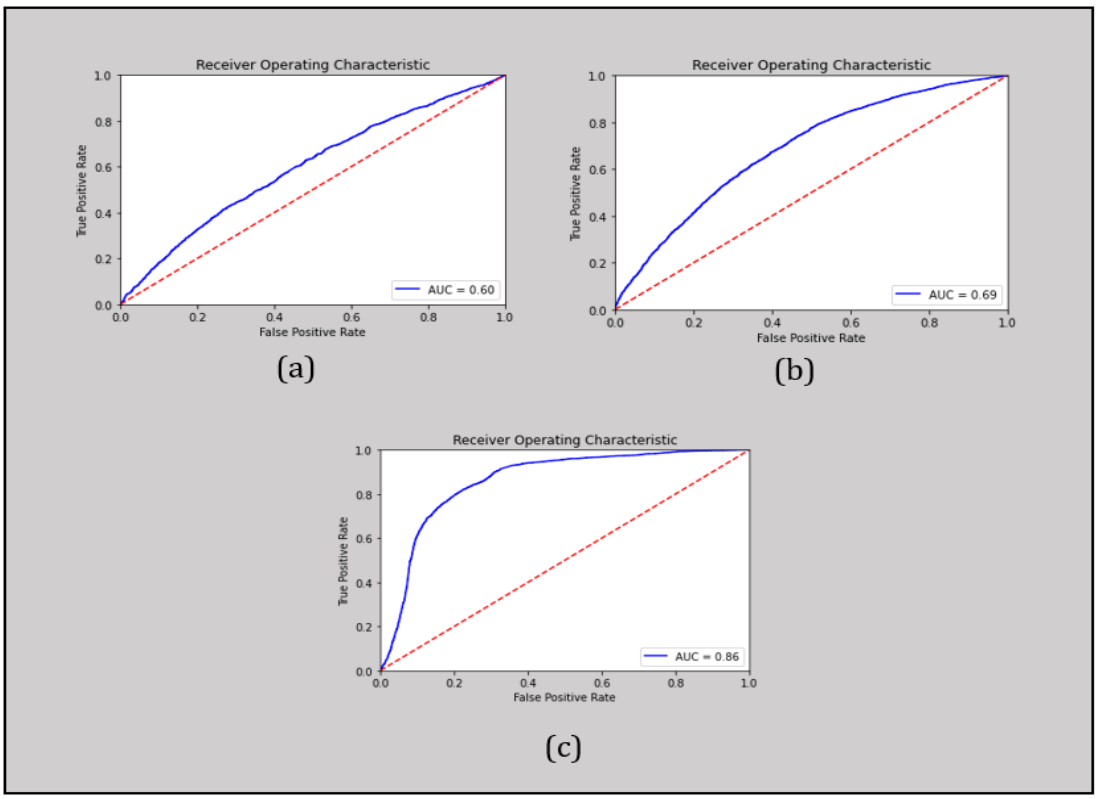


Figura 6 – AUC para o classificador (a) SVM sem OHP (b) LR (c) SVM com OHP

Para referência, Dastile et al. [2020], comparando 39 diferentes estudos que avaliaram a performance de diferentes algoritmos de ML para CS e dividiam-se entre as mesmas duas bases de dados, encontraram uma média próxima a 0.8 para AUC dos modelos SVM, atestando os bons resultados do classificador testado neste trabalho.

6. Conclusão

As diligências tomadas por instituições financeiras no curso do processo de concessão de crédito são vitais para o seu funcionamento e sucesso. Representando a mais importante atividade destas instituições, a concessão de crédito também é vital para o desenvolvimento social e econômico do país. Desta maneira, buscando a coesão do sistema creditício, é importante que sejam tomados esforços para assegurar o adimplemento dos empréstimos na indústria financeira.

Modelos e algoritmos de classificação apresentam possíveis soluções. Apesar de LR ser o padrão na modelagem do risco de crédito pela indústria financeira, inovações na área de ML trouxeram novos algoritmos de classificação com maior sofisticação e complexidade. Realizando a modelagem de CS proposta com a aplicação do algoritmo SVM, este estudo atestou a eficácia desta abordagem, tendo as métricas de avaliação confirmado seu excelente poder discriminatório, significativamente superior ao da abordagem tradicional da LR. Também foi claramente evidenciada a importância do devido tratamento de dados, balanceamento de classes e da etapa de OHP, que devem ser apropriadamente incorporados ao *framework* do processo.

Contudo, o poder de classificação do modelo testado ainda pode ser aprimorado. Para tal, como sugestão para estudos futuros, seria proveitosa a remodelagem do classificador com

diferentes atributos e metodologias de OHP, além do teste de novos modelos embasados em outros algoritmos de ML, especialmente os métodos *ensemble*, que tem demonstrado resultados promissores na literatura [Dastile et al. 2020]. Também seria interessante para a indústria de crédito a realização de estudos acerca da interpretabilidade dos modelos mais sofisticados de ML para CS, visto que, conforme Dumitrescu [2021], dificuldades em sua devida interpretação costumam ser incompatíveis com os requerimentos de agências reguladoras e constituem a maior razão da preferência pela metodologia de LR, que demonstra maior facilidade de implementação e interpretação.

7. Agradecimentos

Os autores agradecem à CAPES, ao CNPq e à FAPEMIG.

Referências

BACH, Małgorzata; WERNER, Aleksandra; PALT, Mateusz. The proposal of undersampling method for learning from imbalanced datasets. **Procedia Computer Science**, v. 159, p. 125-134, 2019.

BAESENS, Bart et al. Benchmarking state-of-the-art classification algorithms for credit scoring. **Journal of the operational research society**, v. 54, n. 6, p. 627-635, 2003.

CAO, Yuzhou; XU, Yitian; DU, Junling. Multi-variable estimation-based safe screening rule for small sphere and large margin support vector machine. **Knowledge-Based Systems**, v. 191, n. xxxx, p. 105223, 2020.

CHEN, Tong. Credit Default Risk Prediction of Lenders with Resampling Methods. In: **2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)**. IEEE, 2021. p. 123-127.

COENEN, Lize; VERBEKE, Wouter; GUNS, Tias. Machine learning methods for short-term probability of default: A comparison of classification, regression and ranking methods. **Journal of the Operational Research Society**, v. 73, n. 1, p. 191-206, 2022.

DASTILE, Xolani; CELIK, Turgay; POTSANE, Moshe. Statistical and machine learning models in credit scoring: A systematic literature survey. **Applied Soft Computing**, v. 91, p. 106263, 2020.

DUMITRESCU, Elena-Ivona et al. Machine Learning or Econometrics for Credit Scoring: Let's Get the Best of Both Worlds. 2021.

GHORBANI, Ramin; GHOUSI, Rouzbeh. Comparing different resampling methods in predicting students' performance using machine learning techniques. **IEEE Access**, v. 8, p. 67899-67911, 2020.

JENG, Jin Tsong. Hybrid approach of selecting hyperparameters of support vector machine for regression. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics**, v. 36, n. 3, p. 699–709, 2006.

LESSMANN, Stefan et al. Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. **European Journal of Operational Research**, v. 247, n. 1, p. 124-136, 2015.

LI, Y. F. *et al.* A systematic comparison of metamodeling techniques for simulation optimization in Decision Support Systems. **Applied Soft Computing Journal**, v. 10, n. 4, p. 1257–1273, 2010.

MARQUES, A. I.; GARCÍA, Vicente; SÁNCHEZ, José Salvador. A literature review on the application of evolutionary computing to credit scoring. **Journal of the Operational Research Society**, v. 64, n. 9, p. 1384-1399, 2013.

MEYER-BAESE, Anke; SCHMID, Volker J. **Pattern recognition and signal analysis in medical imaging**. Elsevier, 2014.

NIKLIS, Dimitrios; DOUMPOS, Michael; ZOPOUNIDIS, Constantin. Combining market and accounting-based models for credit scoring using a classification scheme based on support vector machines. **Applied Mathematics and Computation**, v. 234, p. 69-81, 2014.

OLIVEIRA, Adriano L.I. *et al***.** GA-based method for feature selection and parameters optimization for machine learning regression applied to software effort estimation. **Information and Software Technology**, v. 52, n. 11, p. 1155–1166, 2010.

ONEL, Melis et al. Big data approach to batch process monitoring: Simultaneous fault detection and diagnosis using nonlinear support vector machine-based feature selection. **Computers & chemical engineering**, v. 115, p. 46-63, 2018.

ØSTERGÅRD, Torben; JENSEN, Rasmus Lund; MAAGAARD, Steffen Enersen. A comparison of six metamodeling techniques applied to building performance simulations. **Applied Energy**, v. 211, n. October 2017, p. 89–103, 2018.

RANI, Alka et al. Machine learning for soil moisture assessment. In: **Deep Learning for Sustainable Agriculture**. Academic Press, 2022. p. 143-168.

SHEN, Feng; WANG, Run; SHEN, Yu. A cost-sensitive logistic regression credit scoring model based on multi-objective optimization approach. **Technological and Economic Development of Economy**, v. 26, n. 2, p. 405-429, 2020.

TRUCCO, Emanuele; MACGILLIVRAY, Tom; XU, Yanwu (Ed.). **Computational retinal image analysis: Tools, applications and perspectives**. Academic Press, 2019.

VAPNIK, Vladimir N. **The Nature of Statistical Learning Theory**. New York: [s.n.], 1999.

WANG, Yuelin et al. A Comparative Assessment of Credit Risk Model Based on Machine Learning——a case study of bank loan data. **Procedia Computer Science**, v. 174, p. 141-149, 2020.

WU, Jia et al. Hyperparameter optimization for machine learning models based on Bayesian optimization. **Journal of Electronic Science and Technology**, v. 17, n. 1, p. 26-40, 2019.

YANG, Shengping; BERDINE, Gilbert. The receiver operating characteristic (ROC) curve. **The Southwest Respiratory and Critical Care Chronicles**, v. 5, n. 19, p. 34-36, 2017.
Wu et al 2019