UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ADMINISTRAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM ADMINISTRAÇÃO

VICTOR GOMES HELDER

COMPARAÇÃO DE TÉCNICAS DE MACHINE LEARNING PARA PREDIÇÃO DE DEFAULT E APLICAÇÃO DA HEURÍSTICA VNS PARA SELEÇÃO DE VARIÁVEIS

PORTO ALEGRE
2021

VICTOR GOMES HELDER

COMPARAÇÃO DE TÉCNICAS DE MACHINE LEARNING PARA PREDIÇÃO DE DEFAULT E APLICAÇÃO DA HEURÍSTICA VNS PARA SELEÇÃO DE VARIÁVEIS

Dissertação de Mestrado Acadêmico apresentada ao Programa de Pós-Graduação em Administração da Universidade Federal do Rio Grande do Sul como requisito à obtenção de Grau de Mestre em Administração.

Orientador: Prof. Dr. Tiago Pascoal Filomena

PORTO ALEGRE
2021

CIP - Catalogação na Publicação

Helder, Victor Gomes
Comparação de Técnicas de Machine Learning para Predição de Default e Aplicação da Heurística VNS para Seleção de Variáveis / Victor Gomes Helder. -- 2021.
45 f.
Orientador: Tiago Pascoal Filomena.

Dissertação (Mestrado) -- Universidade Federal do Rio Grande do Sul, Escola de Administração, Programa de Pós-Graduação em Administração, Porto Alegre, BR-RS, 2021.

1. Credit Scoring. 2. Machine Learning. 3. Seleção de Variáveis. I. Filomena, Tiago Pascoal, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os dados fornecidos pelo(a) autor(a).

VICTOR GOMES HELDER

COMPARAÇÃO DE TÉCNICAS DE MACHINE LEARNING PARA PREDIÇÃO DE DEFAULT E APLICAÇÃO DA HEURÍSTICA VNS PARA SELEÇÃO DE VARIÁVEIS

Dissertação de Mestrado Acadêmico apresentada ao Programa de Pós-Graduação em Administração da Universidade Federal do Rio Grande do Sul como requisito à obtenção de Grau de Mestre em Administração.

Banca Examinadora:

Prof. Dr. Tiago Pascoal Filomena
PPGA/UFRGS
Orientador

Prof. Dr. Luciano Ferreira
PPGA/UFRGS

Prof. Dr. Guilherme Kirch
PPGA/UFRGS

Prof. Dr. Michel José Anzanello
PPGEP/UFRGS

Porto Alegre, 09 de Junho de 2021

RESUMO

Credit scoring possui um papel fundamental para instituições financeiras no processo de análise para concessão de crédito. Nesse sentido, técnicas de *machine learning* têm sido utilizadas para desenvolver modelos de *credit scoring*, uma vez que elas buscam reconhecer padrões existentes em bases de dados contendo o histórico de tomadores de crédito, e assim podem inferir quais indivíduos terão mais propensão a cometer um calote (*default*). Entretanto, essas bases de dados comumente apresentam um grande número de variáveis, algumas das quais podem ser ruidosas, o que prejudica a análise. No presente trabalho, é proposta uma técnica de seleção de variáveis baseada em um conceito de vizinhança variável, chamado VNS. A aplicabilidade do método é avaliada em conjunto com sete das principais técnicas utilizadas para fazer predição de *default* em problemas de análise de crédito. Seu desempenho foi comparado com a seleção de variáveis obtida pelo conhecido método estatístico PCA. Os resultados indicam performance superior do VNS na maior parte dos testes aplicados, sugerindo a robustez do método.

Palavras-chave: *Credit Scoring*. *Machine Learning*. Seleção de variáveis. VNS - *Variable Neighborhood Search*.

ABSTRACT

Credit scoring plays a major role for financial institutions when making credit-granting decisions. In this context, machine learning techniques have been used to develop a credit scoring model, as they seek to recognize existing patterns in databases containing the credit history of borrowers to infer potential defaulters. However, these databases often contain a large number of variables, some of which can be noisy, leading to imprecise results. In the present work, a feature selection technique is proposed based on a variable neighborhood concept, so-called VNS. The applicability of the method is assessed in conjunction with seven of the main techniques used to make default prediction in credit analysis problems. Its performance was compared to the feature selection obtained by the well-known PCA statistical method. The results indicate superior performance of the VNS in most of the applied tests, suggesting the robustness of the method.

Keywords: Credit Scoring. Machine Learning. Feature Selection. VNS - Variable Neighborhood Search.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

É fato que a concessão de crédito desempenha um importante papel na economia mundial e na sociedade de um modo geral. Muitas vezes, é apenas através de um empréstimo que indivíduos conseguem adquirir bens que desejam, assim como, grande parte dos negócios, só tem início após algum tipo de concessão de crédito. Dessa forma, operações de crédito estão presentes em quase todos os tipos e escalas de transações financeiras.

A atividade de empréstimo é extremamente antiga, com relatos chegando a até 2000 A.C. (THOMAS, 2009). Entretanto, apenas no século XV D.C. configurou-se uma indústria de empréstimo ao consumidor, com casas de penhores estabelecendo-se na Itália medieval. Nos anos seguintes, houve um debate filosófico a respeito da cobrança de taxas pelo empréstimo de dinheiro, pois a noção do valor do tempo ainda não estava estabelecida, então havia o entendimento de que dinheiro não gera dinheiro por si só, logo seria moralmente errado gerar receita apenas pelo seu empréstimo. No século XVII, com o crescimento da classe média, viu-se um aumento no número de bancos privados dispostos a conceder empréstimos, ainda que para uma pequena parte da população. Entretanto, uma significativa popularização do crédito para consumidores em geral, ocorreu após a implantação das linhas de produção de Henry Ford, na década de 1910, que tornaram os automóveis, até então bens de consumo de luxo, acessíveis a grande parte da população, apesar de ainda requerer um certo investimento para sua aquisição (ANDERSON, 2007). A introdução do cartão de crédito nas décadas de 1950 e 1960 proporcionou ao consumidor um produto que permitia o uso de crédito para grande parte de suas compras, configurando um importante passo na sua disseminação.

A participação do crédito na economia atual atinge níveis expressivos. Segundo dados da Agência Brasil, apenas no Brasil, o saldo de empréstimos ofertados pelos bancos chegou a R$ 3,26 trilhões no ano de 2018, o que representa cerca de 47,4% do Produto Interno Bruto (PIB) do país. Desse total, aproximadamente 55% é relativo à empréstimos pessoais, modalidade essa que apresentou crescimento maior do que empréstimos empresariais no período citado.

Devido ao grande volume de capital associado ao crédito, surge a preocupação acerca do pagamento deste pelo indivíduo que tomou o empréstimo. Segundo dados de uma pesquisa realizada pela CNDL e SPC Brasil, 53,4% dos entrevistados declararam que obtiveram empréstimos sem qualquer garantia. Tal dado reforça ainda mais a importância para que a concessão de crédito seja feita de forma altamente criteriosa.

Um dos principais elementos causadores da mais grave crise econômica do século XXI, a crise financeira de 2008, foi a abundante oferta de crédito de alto risco, à bancos, fundos e companhias de seguro através de complexos instrumentos financeiros (SIKKA, 2009), sobretudo com relação ao mercado imobiliário. Nesse sentido, nos anos que precederam a crise, foram atingidos níveis inéditos de concessão de crédito à mutuários considerados de alto risco (ARNER, 2009).

Dado à importância de se conceder crédito de forma responsável, foram desenvolvidos instrumentos para auxiliar a análise dos dados daqueles que requerem um empréstimo. Nesse contexto, o *credit scoring* apresenta-se como um meio utilizado para classificar indivíduos em "bons" pagadores, ou "maus" pagadores, de acordo com sua propensão a quitar seus empréstimos (HAND; HENLEY, 1997). Existe um grande leque de técnicas que podem ser utilizadas para compor um modelo de *credit scoring*, as quais, de maneira geral, baseiam-se na análise e classificação de um banco de dados contendo variadas informações a respeito de mutuários, de forma que, a partir dessa análise, seja possível classificar novos dados nas categorias de interesse. No decorrer do presente trabalho serão apresentadas algumas das principais técnicas, bem como uma comparação entre as mesmas, no sentido de inferir quais delas são capazes de predizer com maior exatidão a qual categoria (*default* ou não *default*) um novo dado pertence. Para reduzir a probabilidade de que os dados analisados provoquem um viés nos resultados, a comparação será realizada através da utilização de 5 diferentes bases de dados.

Em problemas envolvendo análise de crédito e *credit scoring*, é comum deparar-se com bases de dados que possuem um grande número de variáveis relacionadas aos tomadores de empréstimos. Contudo, em muitos casos, uma expressiva quantidade de variáveis pode significar excesso de informação, gerando ruído na base de dados. Nesse contexto, *feature selection*, ou seleção de variáveis, se apresenta como uma forma de simplificar uma base de dados, excluindo variáveis irrelevantes ou redundantes, agilizando assim o processo de análise dos dados, ao mesmo tempo em que melhora a performance do modelo preditor. Há diversas técnicas conhecidas capazes de realizar uma seleção de variáveis, entretanto, a obtenção de um conjunto ótimo exige um maior aprofundamento na questão. Nesse sentido, técnicas de busca meta-heurística são interessantes alternativas, pois estas adotam uma estrutura de busca mais refinada, o que auxilia no alcance de conjuntos de variáveis mais próximos do ótimo.

O uso de heurísticas ou meta-heurísticas aparenta ser uma boa forma de abordagem para o problema de seleção de variáveis, como aponta Chen e Li (2010), entretanto, esse campo de estudo ainda parece ter bastante espaço para aperfeiçoamento. Em um estudo envolvendo a aplicação de uma metodologia que combina algoritmos para seleção de variáveis e de classificadores, Zhang, He e Zhang (2019) obtiveram bons resultados em termos preditivos, ainda assim, os autores apontam para o fato de que o algoritmo utilizado para fazer a *feature selection* pode ser mais eficiente. A fim de contribuir com uma alternativa ainda pouco explorada para realizar seleção de variáveis no segmento de *credit scoring*, no presente trabalho é proposta uma metaheurística baseada em um conceito de variação das vizinhanças (VNS), a qual será utilizada em conjunto com métodos de *machine learning*, com o intuito de se obter uma predição mais eficiente.

1.1 OBJETIVOS

Dadas as questões levantadas, esta dissertação possui dois objetivos principais:
I) Testar e comparar sete dos principais métodos utilizados para fazer previsão de *default* de crédito;
II) Propor uma meta-heurística de *feature selection* para reduzir o número de variáveis analisadas nas bases de dados, e aumentar a capacidade preditiva dos métodos.

1.2 Organização da Dissertação

A seguir, na seção 2, é apresentada uma revisão da literatura de *credit scoring*, bem como de alguns dos principais métodos de *machine learning* utilizados para fazer classificação de mutuários em relação a sua propensão a cometer um calote, além de uma revisão de *feature selection*, especialmente no que diz respeito a sua aplicação no contexto de *credit scoring*. Na seção 3, será descrita a metodologia aplicada no trabalho, compreendendo os critérios de avaliação adotados, as bases de dados utilizadas e a forma como os métodos preditivos e de *feature selection* foram implementados. No item 4 são apresentados os resultados referentes a aplicação da metodologia proposta.

2 REFERENCIAL TEÓRICO

Nesta seção será apresentado o embasamento teórico dos conceitos e métodos utilizados neste trabalho. Inicialmente, são discutidos os conceitos de *credit scoring* e das técnicas de *machine learning* utilizadas para gerar os modelos de predição. Em seguida, é feita uma revisão de *feature selection* e suas aplicações em *credit scoring*.

2.1 Credit Scoring

Credit scoring pode ser definido como um conjunto de modelos de decisão que auxiliam credores no processo de concessão de crédito ao consumidor (THOMAS; CROOK; EDELMAN, 2017). Tradicionalmente, *credit scoring* é usado para estimar o risco de ocorrência de calote (*default risk*). Segundo Thomas (2009), é usual considerar como risco de calote a chance de um indivíduo atrasar por mais de 90 dias um pagamento nos 12 meses seguintes.

A ideia da utilização de *credit scoring* é tornar o processo de decisão consistente e automático (THOMAS, 2009), o que é especialmente útil na presença de um grande volume de requerentes. Ainda segundo o autor, a filosofia por trás do *credit scoring* é pragmática, pois busca-se apenas melhorar o poder de predição, e não explicar os fatores por trás das previsões. Modelos de *credit scoring* reduzem o custo de análise de crédito, contribuem para as decisões de concessão, e salvam tempo e esforço (ONG; HUANG; TZENG, 2005).

É válido mencionar que técnicas de *credit scoring* avaliam o risco de se emprestar dinheiro para determinado requerente, entretanto, não avaliam a credibilidade desse indivíduo (THOMAS; CROOK; EDELMAN, 2017). Nesse contexto, os autores destacam que credibilidade não é uma característica individual como altura ou mesmo renda, mas sim uma avaliação por parte do credor em relação a um mutuário, considerando um provável cenário econômico futuro. Dessa forma, diferentes credores podem ter diferentes conclusões sobre a credibilidade de um mesmo indivíduo.

Existe uma linha de estudo que visa a avaliar a situação do crédito de clientes que já gozam deste, que é o chamado *behavioral bcoring*. Segundo Thomas (2000), as decisões a serem feitas nesse contexto podem incluir alterações no limite de crédito, se deve ou não serem oferecidos novos produtos, ou ainda como gerenciar casos de maus pagadores. O autor também salienta que quando comparado a sistemas de *credit scoring*, *behavioral bcoring* apresentam informações extras, como o reembolso e o histórico de pedidos do consumidor. Diante disso, modelos de *behavioral bcoring* podem ser divididos em duas categorias, aqueles que usam modelos de credit scoring mas com o acréscimo das variáveis anteriormente mencionadas, e aqueles que constroem modelos de probabilidade de comportamento do cliente.

Uma das primeiras metodologias a se aproximar de uma técnica de mensuração de *credit risk* foi apresentada na década de 1970, por Merton (1974). Ele propôs um modelo para avaliar a

probabilidade de uma companhia dar um calote, isto é, o *credit risk* dessa companhia, levando em consideração parâmetros como valor de mercado dos ativos e volatilidade dos ativos. Já na década seguinte, segundo Reichert, Cho e Wagner (1983), o método *Multiple Discriminant Analysis* (MDA) era o mais utilizado modelo estatístico na época do estudo. O método MDA investiga diferenças encontradas em diferentes grupos, então classifica as observações em grupos pré-determinados e identifica as principais variáveis que contribuem para uma maior discriminação entre os grupos.

Atualmente a utilização de *credit scoring* por instituições financeiras está bastante disseminada. Segundo West (2000), na época da publicação do artigo, era estimado que 97% dos bancos que aceitavam solicitações de cartão de crédito utilizavam algum modelo de *credit scoring*. Em termos de pesquisa acadêmica, também é notório que existe um grande interesse na área, nesse aspecto, Huang (2015) aponta para um significativo aumento de pesquisas acerca do tema a partir de 2009, ou seja, logo após a crise imobiliária norte americana de 2008.

São várias as técnicas que podem ser aplicadas para gerar um *credit scoring*, todavia, um ponto vital que todas têm em comum é a análise de uma grande base de dados contendo informações sobre mutuários e seus históricos de empréstimos anteriores (THOMAS; CROOK; EDELMAN, 2017). As técnicas buscam conectar as características dos indivíduos com seus respectivos históricos, geralmente separando-os em "bons" ou "maus" pagadores, ou seja, aqueles que, respectivamente, não deram calote e aqueles que o deram, no período considerado. O sucesso da aplicação de algumas das primeiras técnicas de *credit scoring* atraiu a atenção de acadêmicos e pesquisadores para o desenvolvimento de métodos estatísticos avançados e técnicas de *Machine Learning* (GESTEL; BAESENS, 2008). Nesse contexto, as informações "aprendidas" com as bases de dados servem como referência para a criação de um modelo que prediga o risco de *default* de requerentes futuros.

2.2 Métodos Preditivos

2.2.1 Regressão Logística

Logistic Regression (LR), ou Regressão Logística, é considerado o método padrão quando se trata de *credit scoring* (LESSMANN et al., 2015). O método da Regressão Logística foi desenvolvido especialmente para ser utilizado nas situações em que as variáveis de saída são binárias (COX, 1958). Ao contrário de outras técnicas de regressão, que utilizam o método dos mínimos quadrados na estimação de parâmetros, a LR utiliza o método da máxima verossimilhança (THOMAS; CROOK; EDELMAN, 2017), o qual determina quais os parâmetros que possuem maior probabilidade de produzir os dados observados.

A Regressão Logística tem um funcionamento semelhante a uma Regressão Linear. Em uma Regressão Linear, a função probabilidade de *default* pode ser dada pela equação 1:

$$p = w_0 + \sum_{i=1}^{n} w_i x_i \tag{1}$$

Nessa situação, o lado direito da equação pode assumir qualquer valor entre $-\infty$ e $+\infty$, entretanto, o lado esquerdo da equação é uma função de probabilidade, e deveria assumir um valor entre 0 e 1 (THOMAS, 2000). A transformação logística pode ser utilizada para linearizar a função probabilidade e limitar a estimação das probabilidades do modelo entre 0 e 1 (SIDDIQI, 2012).

Após a transformação logística, a probabilidade da variável de saída ser 1 é dada pela formulação 2:

$$P(Y = 1|X) = \frac{1}{1 + exp(-wx - b)} \tag{2}$$

em que os parâmetros *w* e *b* podem ser estimados pelo método da máxima verossimilhança (equação 3) para maximizar a função log-verossimilhança (BELLOTTI; CROOK, 2009).

$$L(x_1, x_2, ..., x_n|w, b) = \sum_{i=1}^{n} y_i log p_i + (1 + y_i) log(1 - p_i) \tag{3}$$

Em que $p_i = P(y = 1|x_i)$.

2.2.2 k-Nearest Neighbors

O método *k-Nearest Neighbors* (k-NN), que pode ser traduzido como k-Vizinhos Mais Próximos, é uma técnica não paramétrica que pode ser usada em problemas de estimação ou classificação (HENLEY; HAND, 1996), sendo esse último o interesse do presente trabalho.

O algoritmo k-NN classifica um dado através do cômputo dos votos dos seus k vizinhos mais próximos (BROWN; MUES, 2012). Se k = 1, o algoritmo realizará apenas uma comparação com o vizinho mais próximo, caracterizando um NN simples. A escolha do valor de k é altamente dependente da base de dados. Por via de regra, um k elevado tende a reduzir o efeito de ruído, entretanto, a fronteira do classificador se torna menos distinta. Na Figura 1 é mostrado um esquema para classificação de um novo dado entre duas categorias, utilizando um k igual a 5.

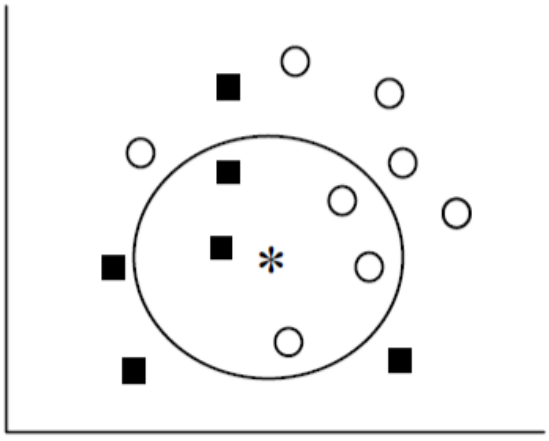

Figura 1 – kNN com k igual a 5 - Adaptado de Hu et al. (2016).

De maneira geral, a distribuição de pesos entre os vizinhos pode ser uniforme ou proporcional à distância até a amostra em análise. No primeiro caso, dado um determinado valor de k, o resultado final será dado simplesmente pela soma dos votos dos k vizinhos mais próximos. Já na segunda situação, Geler et al. (2016) comenta que são atribuídos pesos aos vizinhos de acordo com a distância da amostra a ser classificada, nesse caso, serão atribuídos pesos maiores aos vizinhos mais próximos.

Chomboon et al. (2015) estudou a performance de onze diferentes métricas de distância, e concluiu que a distância Euclidiana é uma das seis que apresentam melhor desempenho, juntamente com Minkowski, Chebychev, Mahalanobis, City-block e Standarized Euclidean. Segundo Hu et al. (2016) a distância Euclidiana é a métrica de distância mais utilizada com o método k-NN. Ainda segundo o autor, a distância Euclidiana normalizada pode ser dada por:

$$dist(A,B) = \sqrt{\frac{\sum_{i=1}^{m}(x_i - y_i)^2}{m}} \tag{4}$$

em que A e B são vetores de variáveis $A = (x_1, x_2, ..., x_m)$ e $B = (y_1, y_2, ..., y_m)$, e *m* é a dimensionalidade do espaço de variáveis.

2.2.3 Bagging

Bootstrap Aggregattion ou *Bagging* é um algoritmo de aprendizado baseado na diversificação dos dados. Nesse sentido, a diversificação dos dados de treinamento ocorre através da geração de vários subconjuntos de dados, sendo cada um deles gerados randomicamente a partir da base de dados original (ALA'RAJ; ABBOD, 2015). A cada iteração, o *Bagging* redefine o subconjunto de treinamento com substituição, dessa forma, algumas instâncias podem ser evocadas várias vezes, enquanto outras podem ser deixadas de fora (MARQUÉS; GARCÍA; SÁNCHEZ, 2012). Após o processamento de todas as iterações, cada subconjunto treinado fornecerá um voto, dessa forma, através da contagem de todos os votos, a melhor estratégia será escolhida.

Bagging é dito um método *ensemble*, pois trabalha junto a um classificador, no sentido de incrementar a capacidade preditiva deste. Árvore de decisão está entre os principais classificadores utilizados em conjunto com um *ensemble*, pois é um dos que mais se beneficiam dessa união, conseguindo ganhos expressivos em termos de acurácia nas predições (WANG et al., 2011).

Conforme Breiman (1996), uma importante característica para uma boa performance do método é a presença de instabilidade. Nesse contexto, um procedimento pode ser dito instável se pequenas alterações no conjunto de treinamento geram grandes mudanças no preditor em aprendizagem. Desse modo, o uso do *Bagging* é particularmente indicado para problemas que apresentam elevado ruído. Dietterich (2000), em seus experimentos, observou que o *Bagging* apresentou resultados bastante satisfatórios na presença de ruído, e isso se deve principalmente ao fato do método explorar instabilidades para produzir classificadores diversos, o que contribui para o aumento da acurácia.

2.2.4 Boosting

Boosting é um método *ensemble* que visa a melhorar a performance de um algoritmo de aprendizado (FREUND; SCHAPIRE et al., 1996). Trata-se de uma estratégia de reamostragem, com uma distribuição de probabilidade que é dependente da taxa de erros de classificação para cada observação (WEST; DELLANA; QIAN, 2005). *Boosting* emprega um algoritmo iterativo que constrói um conjunto de classificadores ao sequencialmente treinar cada membro desse conjunto com subconjuntos de treinamento únicos derivados a partir da base de dados, que aumentam a proeminência de certos exemplos considerados difíceis de ensinar, os quais foram erroneamente classificados por membros dos conjuntos anteriores.

Boosting é similar ao *Bagging* no sentido de criar um conjunto de classificadores ao fazer uma reamostragem da base de dados original (MARQUÉS; GARCÍA; SÁNCHEZ, 2012). Entretanto, diferente do *Bagging*, *Boosting* busca forçar o algoritmo de aprendizagem fraco a modificar seus prognósticos, alterando a distribuição sobre os exemplos de treinamento em função dos erros cometidos por hipóteses geradas anteriormente (FREUND; SCHAPIRE et al., 1996).

AdaBoost (*Adaptative Boosting*) é o algoritmo mais conhecido da família *Boosting* (MARQUÉS; GARCÍA; SÁNCHEZ, 2012). AdaBoost não possui elementos randômicos, e gera um conjunto de árvores ao fazer sucessivos rebalanços dos pesos dos dados de treinamento, sendo que os pesos de cada iteração dependem do histórico das iterações anteriores (BREIMAN, 2001). Inicialmente ocorre uma distribuição igualitária de pesos para todas as instâncias de treinamento, e a cada iteração esses pesos são ajustados de acordo com os erros de classificação obtidos pelos resultados do classificador base (MARQUÉS; GARCÍA; SÁNCHEZ, 2012).

Seguindo a metodologia proposta por Freund, Schapire et al. (1996), o método AdaBoost é iniciado definindo-se o número m de instâncias, e em seguida é criado um subconjunto de dados $S = [(x_1, y_1), ..., (x_m, y_m)]$, em que x_i representa uma instância do espaço X, correspondente ao vetor de atributos, e $y_i \in Y$ é a classe associada a x_i. O classificador é então evocado e treinado com base no subconjunto S, sendo a cada instância atribuída um peso $D_t = 1/m$, em que o subíndice t representa o número iterações. Adotando-se a hipótese inicial $h_t : X \to Y$, calcula-se o erro de h_t através da fórmula 5:

$$\epsilon_t = \sum_{i:h_t(x_i)\neq y_i} D_t(i) \tag{5}$$

O objetivo do método é encontrar uma hipótese que resulte na redução do erro ϵ_t. Se o erro for maior do que $1/2$, o processo é abortado, caso contrário, obtém-se o termo de ajuste dos pesos $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$. A seguir, os pesos são atualizados através da expressão 6:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \times \begin{cases} \beta_t & \text{se} \quad h_t(i) = y_i \\ 1 & \text{caso contrário} \end{cases} \tag{6}$$

em que Z_t é uma constante de normalização. O processo é repetido até que o número máximo de iterações seja atingido. Nesse momento, é gerada a hipótese final, como mostra a equação 7, a

qual é composta pelos votos ponderados das hipóteses anteriores, de forma que os maiores pesos são atribuídos às hipóteses com menor erro.

$$h_{fin}(x) = \arg\max[y \in Y] \sum_{t:h_t(x)=y} log\frac{1}{\beta_t} \tag{7}$$

2.2.5 Random Forest

Random Forest (RF) consiste em um procedimento em que várias árvores de decisão são geradas, cada uma delas contribuindo com um voto, sendo o resultado final obtido através da soma dos votos (BREIMAN, 2001).

Uma árvore de decisão organiza o conhecimento extraído dos dados em uma estrutura hierárquica recursiva composta por nós e galhos (QUINLAN, 1986). Cada nó interno representa um atributo, e é associado com um teste de relevância para classificação de dados. As folhas das árvores correspondem às classes, e os galhos representam cada uma das possibilidades de resultados dos testes aplicados. Um novo exemplo pode ser classificado seguindo os nós e galhos de maneira adequada até que uma folha (classe) seja atingida (LORENA et al., 2011). A cada nó, uma árvore de decisão leva em consideração um termo de impureza, que muitas vezes pode ser a Impureza Gini. Géron (2017) define que um nó é "puro" se todas as instâncias analisadas no treinamento pertencem a mesma classe, situação a qual o coeficiente gini assume um valor nulo. A equação 8 mostra o cálculo do coeficiente gini para o iésimo nó.

$$G_i = 1 - \sum_{k=1}^{n} p_{i,k}^2 \tag{8}$$

Em que $p_{i,k}$ é a proporção de instâncias k entre as instâncias de treinamento no iésimo nó.

Segundo Lorena et al. (2011), para cada árvore, um novo conjunto de treinamento é obtido rearranjando-se randomicamente o conjunto anterior com substituição *bootstrap sampling*. Para determinar o nó de divisão da árvore, um subconjunto com menos atributos é aleatoriamente escolhido. A melhor divisão desses atributos selecionados é então utilizada. A técnica *Random Forest* possui dois parâmetros principais que requerem *tuning*, sendo o número de árvores e o número de atributos para crescer cada árvore (BROWN; MUES, 2012).

Conforme Breiman (2001), no método *Random Forest* não ocorre *overfit* a medida que mais árvores são adicionadas, e Zhang et al. (2017) complementam que o método não apresenta problemas na presença de variáveis categóricas, dados desbalanceados, ou valores faltantes.

2.2.6 Support Vector Machine

O método *Support Vector Machine* (SVM) consiste basicamente em um modelo classificador no qual dados binários são separados por um hiperplano, de forma que a margem entre o hiperplano e os exemplos é maximizada (BELLOTTI; CROOK, 2009).

Quando todos os dados não podem serem separados sem que ocorram alguns erros de classificação, o método buscará separar os dados de treino de forma a minimizar o número de

erros (CORTES; VAPNIK, 1995). Busca-se excluir um pequeno número de dados de forma que o restante possa ser perfeitamente separado por um hiperplano com a maior margem possível. A Figura 2 mostra esquematicamente a separação de duas classes de resultados por um hiperplano linear e sua margem ótima.

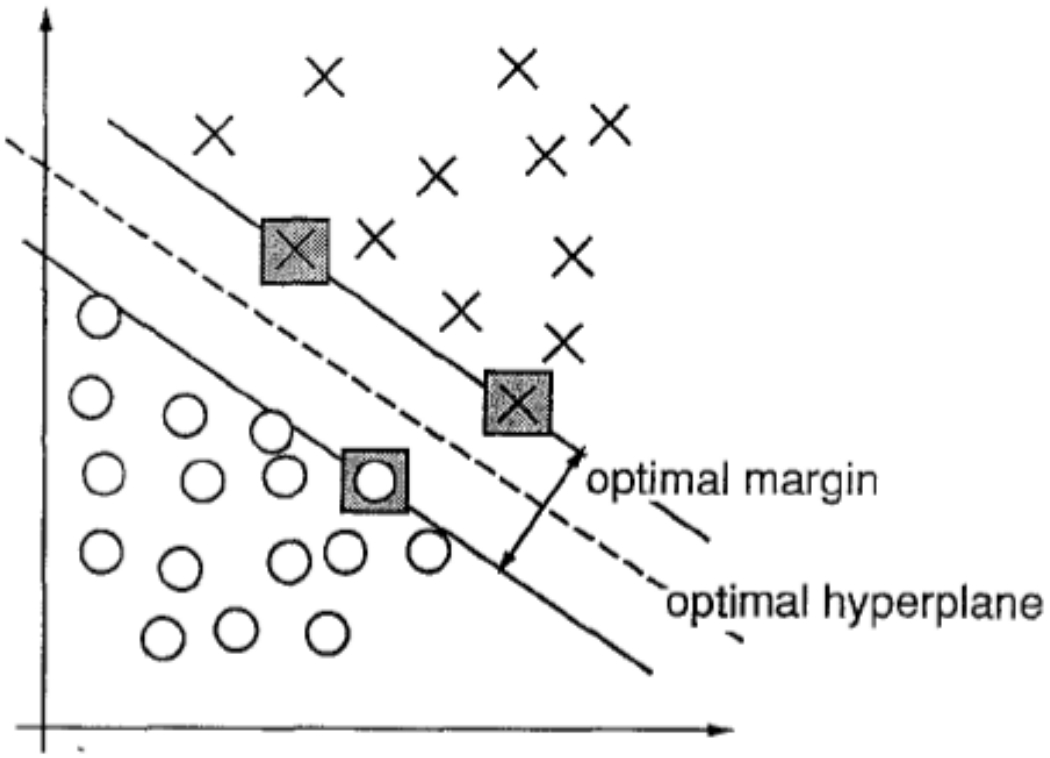

Figura 2 – Hiperplano linear separando duas categorias - Adaptado de Cortes e Vapnik (1995).

Segundo Bellotti e Crook (2009), sendo $y_i \in (-1, +1)$ para todo i=1 até n, o problema de otimização do SVM pode ser representado pela expressão 9:

$$\sum_{i=1}^{n} \alpha_i + \frac{1}{2} \sum_{i,j=1}^{n} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \tag{9}$$

sujeito às restrições 10 e 11:

$$0 \leq \alpha_i \leq C \quad \text{para todo} \quad i = 1 \quad \text{até} \quad n \tag{10}$$

$$\sum_{i=1}^{n} y_i \alpha_i = 0 \tag{11}$$

em que α_i é o multiplicador Lagrangiano para cada exemplo de treinamento i.

O parâmetro C é um termo de custo do problema de classificação. Ele representa uma penalidade usada no SVM em casos de classificações errôneas (BHATTACHARYYA et al., 2011). Nesse sentido, quanto mais alto é o valor de C, mais complexa é a função de classificação.

Uma importante propriedade do método SVM é a representação kernel. Uma função kernel é um recurso especialmente útil quando se trabalha com dados que não são linearmente separáveis. A função de classificação usada no SVM pode ser expressa em termos de produtos escalares, desse modo, utiliza-se uma função kernel para fazer um mapeamento dos dados de entrada para um espaço dimensional de ordem mais alta (BHATTACHARYYA et al., 2011), convertendo assim um problema não separável em um problema separável. Bellotti e Crook (2009) apresenta três das mais utilizadas funções kernels:

Modelo Linear: $k(x_i,x_j) = x_i \cdot x_j$

Modelo Polinomial, com grau *d*: $k(x_i,x_j) = (x_i \cdot x_j + 1)^d$

Função Radial gaussiana (RBF): $k(x_i,x_j) = exp\left(-\frac{||x_i - x_j||^2}{2\sigma^2}\right)$

Em relação ao kernel polinomial, Goldberg e Elhadad (2008) mencionam que um kernel polinomial de ordem superior a 2 costuma gerar *overfit* dos dados.

Martens et al. (2007) destacam que o ponto forte do método reside na sua capacidade de modelar não linearidades, processo esse que resulta em uma alta complexidade matemática, entretanto, esse é também o maior ponto fraco do método, pois torna sua compreensibilidade limitada.

2.2.7 Redes Neurais

Uma rede neural artificial (*artificial neural network*) consiste em um número de pequenas unidades de processamento primitivo interligadas por meio de conexões diretas ponderadas. Cada unidade recebe sinais de entrada através de conexões de entrada ponderadas, e responde mandando um sinal para todas as unidades que possuem conexões de saída (MASSON; WANG, 1990). Pode-se fazer uma analogia desse processo com o funcionamento do cérebro humano, no qual as unidades de processamento representam os neurônios, e as conexões ponderadas representam as sinapses. Além disso, Schmidhuber (2015) comenta que também é possível fazer uma relação com grafos de nós ligados por arcos, em que a primeira camada é o conjunto dos dados (nós) de entrada e então são atribuídos pesos à cada um dos arcos.

O campo de aplicação de Redes Neurais é bastante extenso, com pesquisas em áreas como biologia, psicologia, física, engenharia, entre outras (MASSON; WANG, 1990). Um dos primeiros trabalhos a utilizar Redes Neurais em problemas de *credit scoring*, e comparar com outros métodos de classificação, foi Hand e Henley (1997), nesse contexto, os autores apontaram que Redes Neurais são adequadas em situações nas quais há pouco entendimento acerca da estrutura de dados.

No contexto de *credit scoring*, um modelo de Rede Neural começa passando as variáveis de cada candidato para a camada de entrada, a seguir processa essas variáveis através das camadas escondidas para então atingir a camada de saída. Essa saída é comparada com os resultados desejados, e a partir desse ponto, começa o processo de ajuste dos pesos, os quais numericamente representam as conexões entre os nêutrons. O processo de ajuste dos pesos é realizado repetidas vezes, visando a minimizar os erros entre as previsões e os resultados verdadeiros (MALHOTRA;

MALHOTRA, 2003).

As primeiras arquiteturas de Redes Neurais surgiram na década de 1940, entretanto, os modelos iniciais eram essencialmente variações de métodos de regressão linear, que remontam dos anos de 1800 (SCHMIDHUBER, 2015). Com várias décadas de desenvolvimento, as Redes Neurais diversificaram-se conside- ravelmente, de forma que hoje existem diversas maneiras de implementá-las. Uma das arquiteturas mais comuns é a *Multi-Layer Perceptron* (MLP), que consiste em uma camada de entrada, uma ou mais camadas escondidas e uma camada de saída (BAO; LIANJU; YUE, 2019).

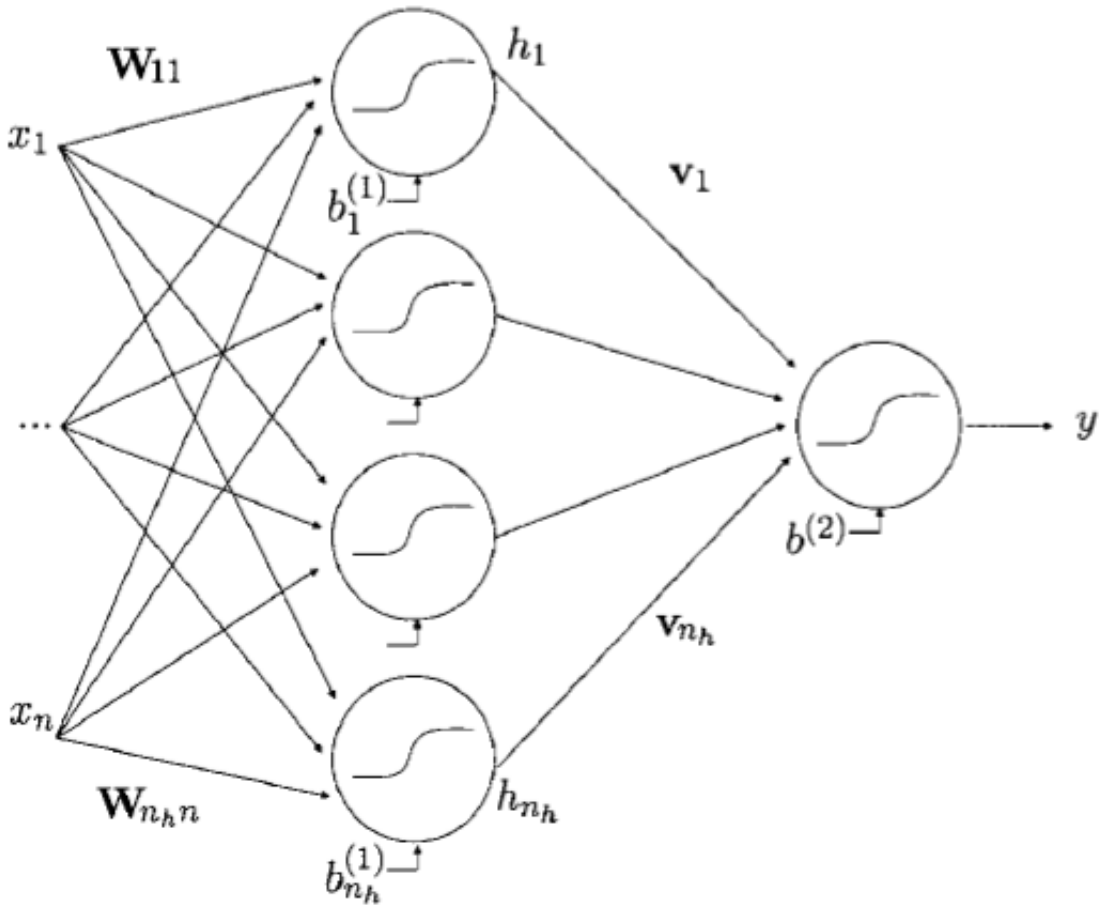

Figura 3 – Arquitetura MLP com uma camada escondida - Adaptado de Baesens et al. (2003)

A Figura 3 mostra esquematicamente uma arquitetura MLP com uma camada escondida e um nurônio como saída. Conforme Baesens et al. (2003), seja $x_j \in J = (1, 2, ..., n)$ o termo que representa os neurônios da camada de entrada, a saída do neurônio h_i da camada escondida será computada pelo processamento dos pesos w_{ij}, que conectam o termo j da camada de entrada com o termo i da camada escondida, e o termo de viés $b_i^{(1)}$, como mostra a equação 12:

$$h_i = f^{(1)} \left(b_i^{(1)} + \sum_{j=1}^{n} W_{ij} x_j \right) \tag{12}$$

Seguindo a mesma lógica, a saída da última camada pode ser obtida da seguinte forma:

$$y = f^{(2)}\left(b^{(1)} + \sum_{j=1}^{n_h} v_j h_j\right) \tag{13}$$

sendo n_h o número de neurônios da camada escondida e v_j os pesos que conectam a camada escondida ao neurônio de saída. As funções de transferência $f^{(1)}$ e $f^{(2)}$ permitem a rede modelar relações não lineares dos dados. Baesens et al. (2003) ainda comentam que em problemas de classificação binários, como é o caso deste trabalho, é recomendado usar uma função sigmoid, como mostra a equação 14.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{14}$$

2.2.8 Aplicações em Credit Scoring

Brown e Mues (2012) realizaram um dos principais comparativos envolvendo a aplicação de métodos preditivos em *credit scoring*. O principal objetivo do estudo consistia em verificar o desempenho de diferentes classificadores na presença de bases de dados desbalanceadas, o que é bastante comum no contexto de *credit scoring*, pois a ocorrência de *defaults* costuma ser consideravelmente menor que a de não *defaults*. Fez-se a análise dos seguintes métodos: Regressão Logística (LR), Análise Discriminante Linear e Quadrática, Árvore de decisão C4.5, k-NN, *Gradient Boosting*, *Random Forest*, SVM e Redes Neurais. Foram utilizadas 5 bases de dados, das quais duas, *Australian* e *German*, também constam no presente trabalho. A fim de verificar o desempenho dos classificadores conforme a proporção de *defaults* presente nos dados, foram testados diferentes níveis de desbalanceamento, variando de 1% até 30% de *defaults* em relação ao total. A medida de avaliação adotada pelos autores foi a *Area under the receiver operator characteristic curve* (AUC), e na Tabela 1 são mostrados os resultados obtidos com os métodos comuns aos dois trabalhos, nas bases de dados *Australian* e *German*, com a presença de 30% de *defaults*, pois é a distribuição mais próxima ao original dessas bases. Os métodos *Random Forest*, *Boosting* e SVM apresentaram os melhores desempenhos de maneira geral, entretanto, os autores apontam que o último não mantém bons resultados quando os dados são extremante desbalanceados, isto é, quando a proporção de *defaults* é muito pequena em relação ao total da amostra.

	Australian	*German*
LR	0,906	0,767
k-NN	0,928	0,750
Grad. Boosting	0,949	0,772
Random Forest	0,937	0,800
SVM	0,951	0,819
Redes Neurais	0,921	0,727

Tabela 1 – Resultados (AUC) de Brown e Mues (2012) para as bases *Australian* e *German*, considerando 30% de *default*.

Marqués, García e Sánchez (2012) concentraram seus estudos na comparação de métodos *ensamble* para fazer predição de *default* em *credit scoring*. Nesse sentido, foram analisados cinco *ensambles*: *Boosting* (AdaBoost), *Bagging*, *Random subspace*, DECORATE e *Rotation forest*. Cada um destes foi utilizado em conjunto com cinco classificadores, os quais são *1-Nearest Neighbour* (1-NN), *Naive Bayes Classifier* (NBC), Regressão Logística (LR), Rede Neural (Multilayer Perceptron – MLP), *Radial Basis Function* (RBF), *Support Vector Machine* (SVM) e Árvore de Decisão (C4.5). As bases de dados utilizadas foram *Australian*, *German*, *Japanese*, *Iranian*, *Polish* e UCDS, sendo que as três primeiras também estão no presente trabalho. O critério de comparação adotado foi a acurácia, em conjunto com os erros de tipo I (taxa de maus aplicantes catalogados como bons) e tipo II (taxa de bons pagadores previstos como maus), entretanto, apenas com relação a acurácia são mostrados os resultados completos para todos os métodos. Alguns dos resultados obtidos pelos autores são mostrados na Tabela 2, a qual apresenta apenas as bases de dados, classificadores e ensambles que também foram utilizados no presente trabalho.

	Australian	*German*	*Japanese*
LR	0,8493	0,7570	0,8729
1-NN	0,8145	0,7050	0,7948
Bagging(c/ C4.5)	0,8594	0,7370	0,8683
Boosting(c/ C4.5)	0,8290	0,8230	8591
SVM	0,8507	0,7600	0,8637
Redes Neurais	0,8304	0,7240	0,8330

Tabela 2 – Resultados (Acurácia) de Marqués, Garcia e Sánchez (2012) para as bases de dados *Australian*, *German* e *Japanese*.

Um dos mais amplos estudos na linha de pesquisa em questão foi conduzido por Lessmann et al. (2015), o qual foi proposto no sentido de fazer uma atualização de Baesens et al. (2003), *benchmark* no segmento. Foi considerado uma série de classificadores, os quais foram testados em oito diferentes bases de dados, entre elas a *Australian Credit Approval* e a *German Credit Data*. Os autores mencionam que o desbalanceamento presente nas bases foi mantido com o intuito de verificar a robustez dos métodos. Foram utilizadas seis medidas para avaliação, entre elas a porcentagem de classificações corretas (PCC), que possui o mesmo sentido que a acurácia, AUC, *partial Gini index* (PG), *H-measure*, *Brier Score* (BS) e *Kolmogorov-Smirnov statistic* (KS). Ao final do estudo, os autores indicam o método *Random Forest* como *benchmark* para ser comparado a futuros novos métodos que venham a surgir, pois apresenta um dos melhores desempenhos a custos computacional e de implementação relativamente baixos.

2.3 Feature Selection

Feature selection, ou seleção de variáveis, é um dos problemas mais fundamentais no campo de *machine learning* (WANG et al., 2012). Seu principal objetivo é determinar

um conjunto mínimo de atributos para representar o problema, enquanto mantém informação suficiente para que a capacidade preditiva não seja prejudicada.

Seleção de variáveis é um importante tópico a ser abordado na construção de sistemas de classificação. Bases de dados de crédito frequentemente apresentam um grande número de variáveis, o que pode levar a um modelo com excessiva complexidade, mas sem uma notável acurácia (WANG et al., 2012). É vantajoso limitar o número de variáveis de entrada em um classificador para se obter um bom preditor e um modelo computacionalmente menos exigente (CHEN; LI, 2010).

Algoritmos de seleção de variáveis podem ser classificados basicamente em duas categorias: *filter approach* e *wrapper approach* (CHEN; LI, 2010). A *filter approach* seleciona um conjunto de variáveis independente do classificador, de modo que a filtragem é baseada em características observadas na fase de treinamento. Já, na *wrapper approach*, geralmente utiliza-se um critério de avaliação associado a um pré-determinado algoritmo de aprendizado para a determinação dos conjuntos a serem selecionados. Chen e Li (2010) ainda comentam que métodos *wrapper* costumam obter melhores resultados em encontrar conjuntos com as variáveis mais relevantes, o que é natural visto que os métodos *wrapper* atuam em conjunto com o classificador. Também é possível combinar as abordagens *filter* e *wrapper*, como apresentado por Beuren e Anzanello (2019), em que a técnica *Mutual Information* é utilizada para remover variáveis menos significativas (na fase de filtragem), e em seguida, três testes não paramétricos (Anderson-Darling, Kruskal-Wallis and Steel's Test) são aplicados para ranquear as variáveis remanescentes (fase *wrapper*).

Muitos métodos *wrapper* são baseados em técnicas de busca heurística. Uma heurística pode ser definida como uma técnica que busca encontrar uma boa solução (ótimo local) a um razoável custo computacional, mas sem garantir o alcance de uma solução ótima (ótimo global) (EL-SHERBENY, 2010). A Figura 4 mostra graficamente o conceito mencionado, onde podem ser observados dois pontos de ótimo local e o ponto de ótimo global. A busca por uma solução ou um conjunto ótimo envolve, de maneira geral, a aplicação de técnicas que possibilitem que o algoritmo utilizado continue buscando por alternativas de solução mesmo após atingir um ponto de ótimo local. Nesse contexto, técnicas de meta-heurística surgem como opções interessantes de resolução. Segundo El-Sherbeny (2010), meta-heurística trata-se de uma estratégia iterativa que guia e modifica as operações das heurísticas subordinadas ao combinar de maneira inteligente diferentes conceitos para explorar o espaço de pesquisa.

Uma meta-heurística ainda pouco explorada para seleção de variáveis em problemas de *credit scoring* é a *Variable Neighborhood Search* (VNS) o que pode ser traduzido para Busca em Vizinhança Variável. Segundo Talbi (2009), a ideia básica do VNS é explorar um conjunto pré definido de vizinhanças para chegar na melhor solução. O método pode explorar aleatoriamente ou sistematicamente o conjunto de vizinhanças afim de atingir diferentes ótimos locais, dessa forma o VNS se vale da prerrogativa de que utilizando diversas vizinhanças, uma delas eventualmente pode conter o ótimo global.

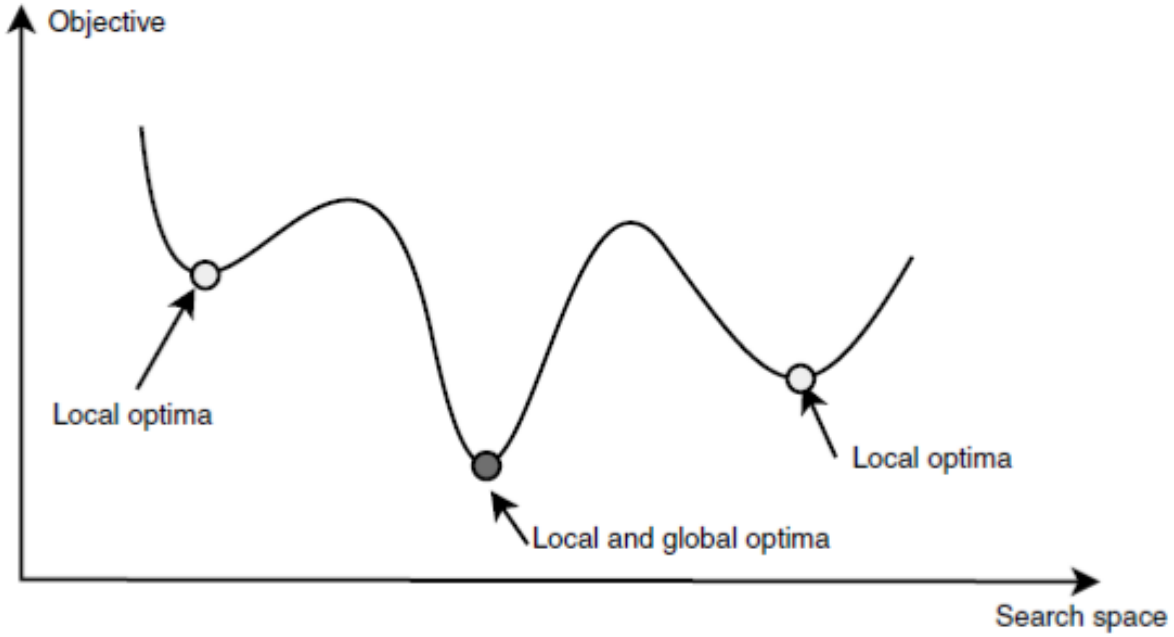

Figura 4 – Pontos de ótimo local e global - Adaptado de Talbi (2009).

O VNS foi proposto por Mladenović e Hansen (1997), como uma meta-heurística alternativa baseada no conceito de exploração das vizinhanças da solução incumbente. Nesse contexto, uma solução incumbente é obtida através de uma busca local utilizando uma primeira estrutura. A seguir o método realiza uma nova busca utilizando outra estrutura de vizinhança. Se a nova solução gerada for melhor que a solução incumbente, o processo é atualizado e a nova solução passa a ser a incumbente, caso contrário, a primeira solução permanece como referência e uma nova busca, utilizando outra estrutura, é realizada. Tal processamento é executado até ser atingida uma condição de parada, que pode ser um determinado número de iterações, tempo de execução, entre outros.

Segundo Talbi (2009), deve ser encontrado um compromisso entre a intensificação da busca e sua diversificação através da distribuição de trabalho entre a fase de busca local e a fase de perturbação, que é onde ocorre a troca de vizinhança. O autor ainda complementa que um VNS será efetivo se as diferentes vizinhanças utilizadas forem complementares no sentido de que os ótimos locais não sejam os mesmos.

2.3.1 Aplicação de Feature Selection em Credit Scoring

Um comparativo envolvendo a aplicação de quatro abordagens de *feature selection* foi apresentado por Chen e Li (2010). O método SVM foi o classificador utilizado como base, nesse sentido, este foi posto em conjunto com cada um dos métodos de *feature selection*, além de também ter sido empregado sem nenhuma seleção de variáveis prévia, para servir como referência. Foram utilizadas as conhecidas bases de dados *Australian Credit Approval* e *German*

Credit Data, e a acurácia e AUC como as medidas de comparação adotadas. Não foi possível apontar um dos quatro métodos (*Linear Discriminat Analysis*, *Rough Sets Theory*, *Decision tree* e *F-score*) como claramente superior no sentido de fazer uma seleção de variáveis, de forma que os autores sugerem o uso de uma heurística para abordar o problema.

Wang et al. (2012) propõem o uso de um método para *feature selection*, denominado pelos autores de RSFS, baseado em *rough sets* e *scatter search*, o qual trata-se de um algoritmo baseado na recombinação de amostras para gerar um grupo de soluções. Esse método foi aplicado em conjunto com três diferentes classificadores, e implementado em duas bases de dados. Os resultados apresentados pelos autores apresentaram ganhos marginais em relação a previsão de *default*, mas houve ganhos significativos no sentido da redução do tempo de processamento.

Zhang, He e Zhang (2019) propuseram a aplicação de uma heurística baseada em um algoritmo genético multipopulacional, no qual vários *filter methods* são combinados para gerar um conjunto ótimo de *features*. Em uma segunda etapa, um algoritmo faz a escolha do classificador que será usado no processo, e depois este é associado a um método *ensemble*. Os testes foram realizados em cinco bases de dados, incluindo as populares *Australian Credit Approval* e *German Credit Data*, e quatro medidas de desempenho foram consideradas, sendo acurácia, AUC, *H measure* e *Brier score*. A metodologia apresentada mostrou bons resultados em termos de predição, entretanto os autores apontam que há espaço para aperfeiçoamento na heurística de seleção de variáveis.

2.3.2 Principal Component Analysis

Com o intuito de avaliar a capacidade preditiva da meta-heurística proposta, optou-se por usar como referência comparativa uma técnica comumente aplicada para efetuar redução de variáveis, o *Principal Component Analysis* (PCA), que é um método estatístico de análise multivariada, recomendado por Song, Guo e Mei (2010) para ser usado em seleção de variáveis. No contexto de *credit scoring* foi utilizado por Šušteršič, Mramor e Zupan (2009) e Han, Han e Zhao (2013), e em um comparativo envolvendo também os métodos de *feature selection Genetic Algorithm*, *Relief Method* e *Information Gain Ration*, Koutanaei, Sajedi e Khanbabaei (2015) considerou o PCA como a melhor escolha.

Segundo Jolliffe (1986), a ideia principal do PCA consiste em reduzir a dimensionalidade de bases de dados contendo variáveis correlacionadas, enquanto retém o máximo de variabilidade dos dados originais. Essa redução é atingida pela transformação dos dados originais em um novo conjunto de variáveis ortogonais, os chamados componentes principais (Principal Components - PCs). Tal processo é obtido pela rotação do sistema de coordenadas original em um novo sistema, de modo que a maior parte da informação relevante é concentrada em torno de um menor número de novos eixos (ŠUŠTERŠIČ; MRAMOR; ZUPAN, 2009). Os componentes principais são ordenados de forma que o primeiro irá manter a maior parcela da variabilidade das variáveis originais, enquanto o segundo componente preservará a segunda maior parcela da variabilidade original, e assim sucessivamente. Cada componente principal é um autovetor da

matriz de variância-covariância das variáveis originais.

Uma das etapas mais sensíveis na implementação do PCA é a determinação do número de componentes principais. Existe uma gama de métodos para a obtenção do número de PCs, entre os quais Uğuz (2011) aponta para o uso do critério da porcentagem acumulada da variância, pois este é relativamente simples de ser implementado e apresenta um desempenho satisfatório. De acordo com esse critério, define-se a porcentagem mínima da variância original a ser preservada, então busca-se obter os n primeiros componentes que somados representem o percentual estabelecido. O valor percentual a ser escolhido é fortemente dependente da base de dados utilizada, entretanto, o autor observa que é comum estabelecer como limite mínimo uma variância de 70% a 90%.

3 METODOLOGIA

Nesta seção será descrito o funcionamento do algoritmo proposto para seleção de variáveis, bem como detalhes acerca da implementação e ajuste dos parâmetros dos métodos preditivos. Posteriormente, são apresentadas as bases de dados utilizadas, e a seguir são discutidos os critérios de comparação adotados.

3.1 VNS

O método apresentado no presente trabalho, baseado no VNS proposto por Mladenović e Hansen (1997), consiste na geração e avaliação de vários conjuntos de *features* junto ao classificador, de forma que, no final do processamento, sejam selecionadas as variáveis que contribuem para a melhor acurácia. O número de variáveis do melhor conjunto é obtido automaticamente durante o processamento do método, e pode variar entre apenas uma até o conjunto total de variáveis.

Inicialmente um conjunto contendo metade das variáveis originais é selecionado e testado junto ao classificador. A seguir, o processo iterativo tem início: a cada iteração é gerado um novo conjunto de variáveis, o qual será originado a partir do conjunto da iteração anterior, porém, o novo grupo perderá uma de suas variáveis, ou receberá uma das variáveis que não estavam selecionadas, com 50% de chance para cada caso. Além disso, é imposta uma restrição para que não sejam gerados grupos já testados, de forma que cada conjunto de variáveis a ser vinculado ao classificador seja único. A seguir, o classificador gera um modelo de predição utilizando o grupo de variáveis incumbente, e os resultados são armazenados. O processo iterativo é finalizado quando um determinado número de iterações é atingido, dessa forma, ao final do processamento, o grupo de variáveis que tiver proporcionado a melhor acurácia ao classificador será selecionado.

3.2 PCA

No presente trabalho, Principal Component Analysis (PCA) foi adotado como principal referência comparativa frente ao VNS. Nesse sentido, a função a ser desempenhada por ambos os métodos é a mesma: reduzir a dimensionalidade do espaço de variáveis. No entanto, ao contrário do VNS, o PCA requer que o número de variáveis a ser mantido seja previamente determinado, o que é feito definindo-se a variância mínima a ser preservada. Como mencionado na seção 2.3.2, é comum estabelecer como limite mínimo uma variância entre 70% e 90% da variância total, desse modo, optou-se por testar, para cada base de dados, três valores variância, sendo eles 70%, 80% e 90%. Na Figura 5 é apresentado o fluxograma dos procedimentos de redução de variáveis propostos.

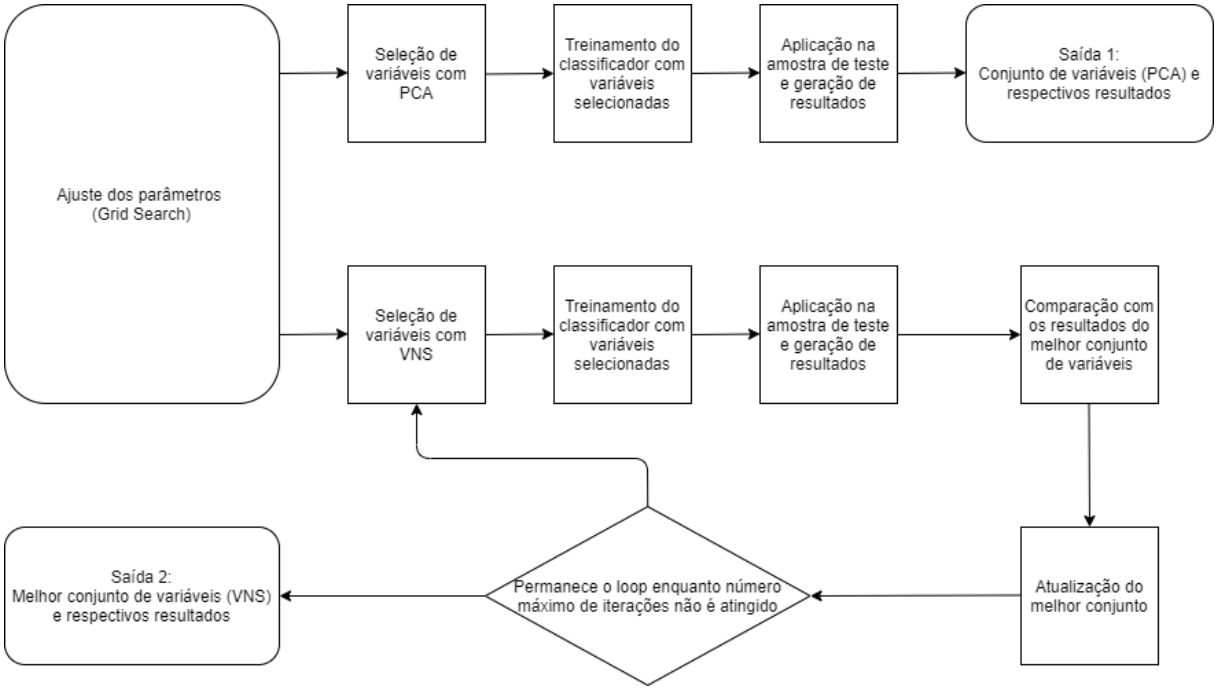

Figura 5 – Fluxograma do VNS proposto.

3.3 Implementação e Ajuste de Parâmetros

O problema foi implementado e desenvolvido na linguagem de programação Python (versão 3.6.8). A implementação dos métodos foi feita através do uso da biblioteca Scikit-Learn, que é uma biblioteca de *Machine Learning* de código aberto, contendo vários algoritmos de classificação e regressão.

Ao se fazer a implementação dos classificadores, a maioria destes requerem ajustes em alguns de seus parâmetros. Os métodos preditivos podem se adaptar de maneira distinta conforme a base de dados utilizada, desse modo, é interessante que seus parâmetros sejam configurados de forma a se atingir a melhor performance possível para um classificador em cada base de dados. Dentre os métodos analisados no presente estudo, a Regressão Logística caracteriza-se como uma exceção no sentido de não possuir parâmetros a serem definidos (BAESENS et al., 2003; BROWN; MUES, 2012).

O método k-NN possui três parâmetros principais a serem definidos durante sua implementação. O princípio básico do método é comparar o dado em análise com seus vizinhos mais próximos, ou seja, instâncias que possuem características semelhantes e já estão classificadas. Diante disso, o número k de vizinhos que serão usados como referência é um parâmetro altamente influente nos resultados, entretanto, possui uma forte dependência em relação a base de dados. Lessmann et al. (2015) recomenda que o valor de k seja um número ímpar, para evitar empates na contabilização dos vizinhos mais próximos. Em relação ao parâmetro seguinte, a métrica de distância, conforme descrito na seção Referencial Teórico, será adotada a distância Euclidiana, por esta ser a mais comum de ser usada em conjunto com o k-NN e apresentar um

dos melhores desempenhos entre as métricas mais conhecidas. Além da métrica de distância, também deve ser definido se serão atribuídos pesos conforme a distância dos vizinhos, ou se todos serão tratados de maneira uniforme.

O método *Random Forest* baseia-se na geração de diversas árvores de decisão, as quais contribuem cada uma com um voto para a geração do modelo definitivo. Seguindo a metodologia adotada por Lessmann et al. (2015), entre os principais parâmetros a serem configurados estão o número de árvores de decisão geradas e a dimensão das subamostras. Os outros *ensambles* abordados neste trabalho, *Bagging* e *Boosting*, também possuem alguns parâmetros a serem definidos. No primeiro caso, conforme Lessmann et al. (2015), foram adotados diferentes valores para o número de subconjuntos gerados. Já no segundo caso, *Boosting*, o parâmetro a ser definido é o número de iterações do modelo.

O método SVM busca separar duas classes de dados através de um hiperplano, de modo que o quanto maior for a margem deste, mais claramente ocorrerá a separação dos dados. A representação kernel é um recurso importante do método SVM, sendo a função responsável por realizar a separação entre as classes de dados, definindo o hiperplano. Conforme mencionado na seção Referencial Teórico, as principais funções kernel são a linear, polinomial e radial gaussiana, e ambas foram consideradas no presente estudo. Além da escolha da função kernel, Thomas, Crook e Edelman (2017) comentam que o parâmetro de penalidade C, ou penalidade de regularização, o qual indica o grau de importância dado a classificações errôneas, deve ser configurado pelo analista. Em relação às Redes Neurais, considerando uma arquitetura MLP, Ala'raj e Abbod (2016) apontam para o *tuning* de dois parâmetros, sendo o número de neurônios na camada escondida e a taxa de aprendizagem. Na Tabela 3 são apresentados de forma sintetizada os parâmetros descritos anteriormente.

Classificador	Parâmetro	Candidatos
Regressão Log.	-	-
kNN	Nro vizinhos (k)	3, 5, 7, 9, ..., 21, 23, 25
	Distribuição dos Pesos	Uniforme, Proporcional
Bagging	Nro Bootstrap Samples (n)	10, 20, 50, 100, 250, 500, 1000
Boosting	Nro iterações (n)	10, 20, 50, 100, 250, 500, 1000
Random Forest	Nro de árvores (n)	100, 250, 500, 750, 1000
	Dimensão subamostra	$\sqrt{m^1}[0.1, 0.25, 0.5, 1, 2, 4]$
SVM	Kernel	Linear, Polinomial, Radial
	Penalidade de regularização (C)	$10^{(-3,-2,...,3)}$
Redes Neurais	Nro neurons hidden l. (n)	10, 50, 100, 250, 500
	Taxa de aprendizado (lr)	0.0005, 0.001, 0.005, 0.01, 0.05

1. m corresponde ao número de variáveis presentes na base de dados.

Tabela 3 – Parâmetros ajustados por método.

3.4 Bases de Dados

Bases de dados de crédito consistem basicamente no agrupamento de determinadas informações sobre indivíduos que obtiveram empréstimos. Nesse contexto, cada instância (linha) representa uma pessoa, enquanto cada coluna indica alguma informação ou característica a respeito dessa pessoa. As informações contidas variam conforme a base de dados, entretanto, um dado imprescindível para o interesse do estudo em questão é o *default*, o qual trata-se de uma variável binária que indica se o indivíduo cometeu algum calote em determinado período. Um valor amplamente aceito para configurar um calote é 90 dias de atraso em um pagamento.

Para este estudo, foram utilizadas cinco bases de dados, sendo todas elas bases públicas encontradas em repositórios digitais. A base Australian Credit Approval (AC) é uma das mais difundidas na literatura de crédito, sendo também uma das mais simples, contando com 14 variáveis de entrada e 690 instâncias. Outra base de dados bastante utilizada é a German Credit Data (GC), a qual possui uma versão contendo apenas variáveis numéricas, resultando em 24 variáveis e 1000 instâncias. Outras bases também consideradas no estudo foram a Japanese Credit Screening Data Set (JC), que contém 689 instâncias e 15 variáveis e a Taiwan Default of Credit Card Clients Data Set (TC), que possui 30000 instâncias e 23 variáveis. Todas as bases acima mencionadas foram obtidas a partir do UCI Machine Learning Repository, de maneira livre. Fez-se também a utilização de uma base de dados obtida a partir da plataforma Kaggle, a Credit Card Data (AER), retirada do livro "Econometric Analysis", a qual possui 1319 instâncias e 11 variáveis. A Tabela 4 apresenta de maneira resumida as principais informações mencionadas acima.

	Nro. de instâncias	Default/não default	Nro. de variáveis	Var. categóricas
AC	690	383/307	14	N
GC (num.)	1000	300/700	24	N
JC	689	383/306	15	S
TC	30000	6636/23364	23	N
AER	1319	296/1023	11	S

Tabela 4 – Principais informações a respeito das bases de dados utilizadas.

3.5 Critérios de Avaliação dos Métodos

A premissa básica do presente trabalho consiste na comparação entre diferentes métodos preditivos utilizados em problemas de *credit scoring*. Nesse sentido, os métodos citados na seção anterior compreendem alguns dos principais mecanismos utilizados pela indústria e por pesquisadores para predizer a probabilidade de um tomador de empréstimo vir a cometer um calote.

Cada uma das técnicas é utilizada em conjunto com cada uma das bases de dados. Nesse contexto, uma porção dos dados, 75%, o que é um valor intermediário em relação ao encontrado

na literatura, é utilizada para fazer o chamado treinamento dos dados, que consiste no processo de relacionar as variáveis de entrada com a variável de saída (*default*), no intuito de criar um modelo, sendo que cada método possui uma maneira particular de gerar o modelo, conforme descrito na seção Referencial Teórico. O modelo gerado é associado aos dados restantes, isto é, os dados de teste, e com base nos padrões obtidos na fase de treinamento, faz uma previsão para a ocorrência de *default*. Os resultados previstos pelo modelo são então comparados com os dados originais. Pelo fato de as respostas serem binárias (ocorreu ou não *default*), quatro possibilidades, as quais compreendem a chamada confusion matrix, como mostra a Tabela 5, emergem da comparação entre as predições e os dados originais, sendo estas verdadeiro positivo (VP) ou negativo (VN), e falso positivo (FP) ou negativo (FN).

	Previsto como bom	Previsto como mau
Bom pagador	VP	FN
Mau pagador	FP	VN

Tabela 5 – *Confusion matrix*.

Existem várias métricas para se comparar o desempenho de diferentes métodos preditivos. Uma das mais elementares e utilizadas trata-se da acurácia, a qual pode ser representada pela fórmula 15:

$$AC = \frac{VP + VN}{VP + VN + FP + FN} \tag{15}$$

Considerando a formulação anterior, percebe-se que a acurácia indica a porcentagem de previsões corretamente realizadas em relação ao total, no entanto, é uma medida um tanto quanto incompleta, no sentido de que não indica se as classificações errôneas possuem um viés, ou seja, apenas com a acurácia não se tem uma percepção caso haja uma predominância de falsos positivos ou falsos negativos.

Duas métricas complementares à acurácia, utilizadas em problemas que envolvem *Machine Learning*, são *Precision* e *Recall*. Conforme Géron (2017), o primeiro representa a porcentagem de não *defaults* corretamente previstos em relação ao total de instâncias previstas como não *default*, ou seja, é a acurácia das previsões positivas. Ainda segundo o autor, o *Recall*, ou sensibilidade, indica a proporção de não *defaults* corretamente previstos em relação ao total de não *defaults*, isto é, a proporção de instâncias positivas corretamente detectadas pelo classificador. Representando-se matematicamente tem-se as equações 16 e 17:

$$Precision = \frac{VP}{VP + FP} \tag{16}$$

$$Recall = \frac{VP}{VP + FN} \tag{17}$$

Outra medida comumente encontrada na literatura é a *Receiver Operating Characteristics* (ROC) *Curve* e sua correspondente *Area Under the Curve* (AUC). O eixo vertical da ROC Curve é chamado de *True Positive Rate* (TPR) ou Taxa de Verdadeiros Positivos, o qual é obtido

da mesma forma que o *Recall*, enquanto o eixo horizontal é denominado *False Positive Rate* (FPR), ou Taxa de Falsos Positivos, e pode ser calculado com a fórmula 18:

$$FPR = \frac{FP}{FP + VN} \tag{18}$$

Existe uma evidente relação entre os termos que compõe a ROC Curve, no sentido de que quanto maior for o *Recall* (ou TPR), mais falsos positivos o classificador irá gerar (GÉRON, 2017). A área situada sob a curva plotada é a medida de maior interesse, e possui uma dimensão entre 0 e 1. Espera-se que um modelo completamente aleatório seja representado por uma linha diagonal ligando o canto inferior esquerdo ao canto superior direito, o que gera uma área de 0,5, indicando que o modelo estará prevendo corretamente cerca de 50% dos testes. Bons preditores possuem uma área mais próxima de 1, deixando a curva ROC mais abaulada em direção ao canto superior esquerdo. Na Figura 6 pode ser visto um exemplo de uma ROC Curve, com a linha tracejada indicando o limiar de um modelo aleatório, e a linha contínua representando uma a curva de um bom preditor.

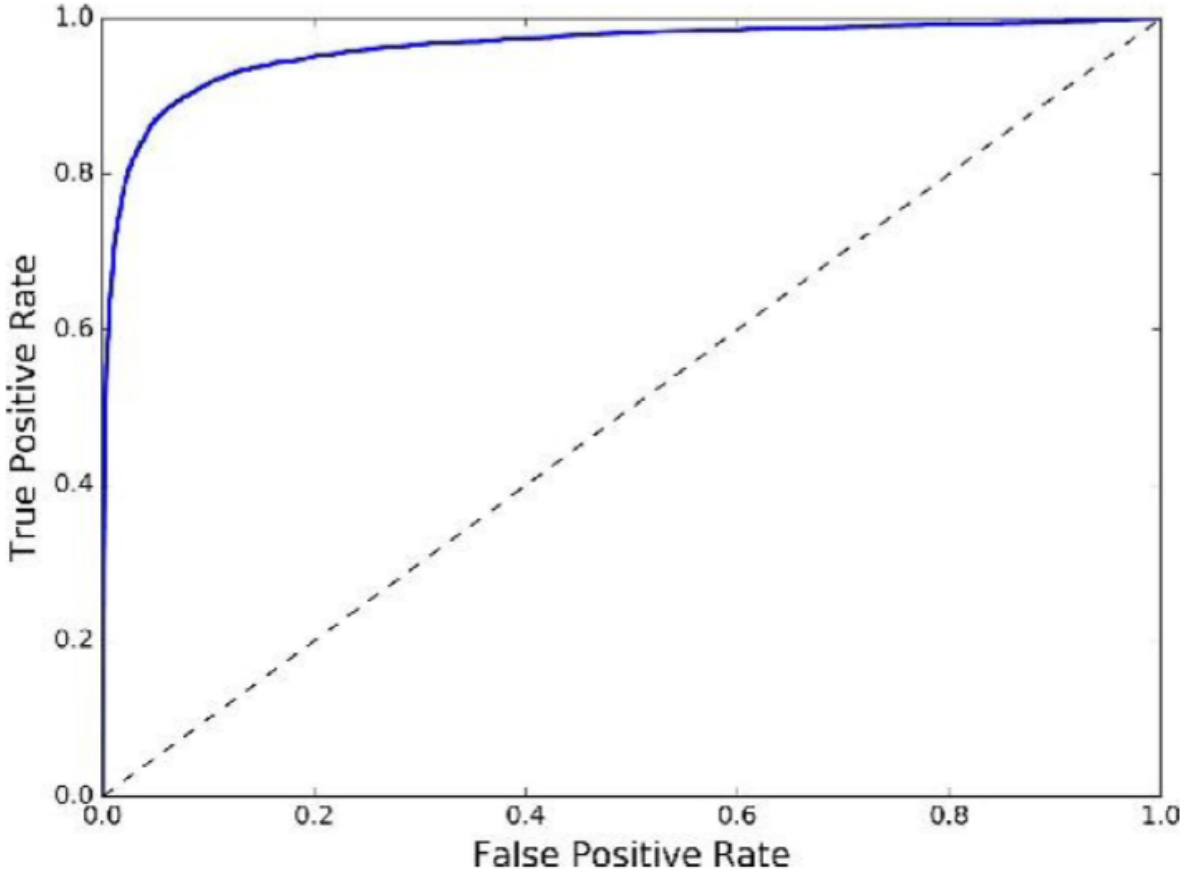

Figura 6 – Exemplo de uma ROC Curve - Adaptado de Géron (2017).

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Nesta seção são apresentados os resultados das predições realizadas em conjunto com os métodos de seleção de variáveis VNS e PCA. A seguir, como referência, também são apresentados os resultados das predições obtidas pelos classificadores considerando a presença de todas as variáveis, isto é, sem a utilização de *feature selection*. Todos os testes foram efetuados utilizando-se um computador com processador *Intel Core i5* de sétima geração, 2,5 GHz e 8 Gb de memória RAM.

4.1 Resultados PCA e VNS

Para a implementação de cada um dos métodos nas bases de dados (à exceção da regressão Logística) foi adotada a técnica *grid search*, comumente encontrada na literatura para a definição dos parâmetros (BELLOTTI; CROOK, 2009; BROWN; MUES, 2012). As Tabelas 6 e 7 mostram de maneira simplificada qual o parâmetro foi considerado para cada combinação entre método e base de dados.

	AC	GC	TC
Regressão Logística	-	-	-
kNN	k=25; Unif.	k=23; Prop.	k=15; Prop.
Bagging	n=100	n=500	n=50
Boosting	n=10	n=50	n=20
Random Forest	n=100; m*0.25	n=1000; m*4	n=100; m*0.5
SVM	Linear; C=0.01	Linear; C=10	Linear; C=0.1
Redes Neurais	n=250; lr=0.005	n=500; lr=0.005	n=10; lr=0.05

Tabela 6 – Parâmetros definidos por método para as bases de dados AC, GC e TC.

	JC	AER
Regressão Logística	-	-
kNN	k=17; Unif.	k=7; Prop.
Bagging	n=20	n=50
Boosting	n=10	n=10
Random Forest	n=100; m*0.5	n=100; m*0.25
SVM	Linear; C=0.01	Linear; C=1000
Redes Neurais	n=250; lr=0.005	n=100; lr=0.01

Tabela 7 – Parâmetros definidos por método para as bases de dados JC e AER.

Uma etapa particularmente sensível a ser implementada está relacionada ao PCA, mais especificamente, quanto a escolha da variância mínima a ser preservada. Para cada base de dados, três níveis de limite mínimo de variância foram estabelecidos, sendo eles 70%, 80% e 90%, de forma que para cada comparação realizada, foi mantido o nível de variância que proporcionou a

melhor performance. A Tabela 8 mostra o número de componentes principais encontrado para cada nível de variância considerado, em cada uma das bases de dados. Convém mencionar que o número de componentes independe do classificador utilizado, uma vez que o PCA, por ser utilizado como *filter*, é aplicado antes da atuação do método de *machine learning*. Em negrito, são indicados os números de componentes selecionadas para cada *dataset*.

	AC	GC	TC	JC	AER
100%	14	24	23	15	11
90%	7	15	**6**	**9**	**6**
80%	**5**	**11**	4	7	4
70%	4	9	3	5	3

Tabela 8 – Número de componentes principais para cada nível de variância considerado.

Pelo fato do método VSN ser iterativo, o número de variáveis selecionadas pode variar a cada teste realizado. A Tabela 9 apresenta o número de variáveis médio selecionado nas vinte aplicações realizadas para cada um dos sete classificadores, em cada base de dados. Pode-se inferir que método SVM tende a reter mais variáveis durante a seleção, uma vez que nas cinco bases analisadas este foi o método que concentrou mais variáveis após a aplicação do VNS. O contrário também pode ser inferido a respeito do k-NN, visto que foi o método a ter mais variáveis descartadas.

	AC	GC	TC	JC	AER
Regressão Logística	10	18	13	12	8
kNN	9	13	7	9	2
Bagging	10	17	20	11	7
Boosting	9	15	10	10	8
Random Forest	10	16	19	10	6
SVM	14	20	22	15	8
Redes Neurais	9	16	15	10	6

Tabela 9 – Número de variáveis médio obtido para cada combinação classificador/base de dados.

Os resultados obtidos após o processamento de cada um dos métodos nas bases de dados AC, GC, TC, JC e AER, em conjunto com PCA e VNS, são mostrados nas Tabelas 10 a 14. A coluna à esquerda indica o classificador base, as quatro colunas a seguir representam a acurácia, precision, recall e AUC obtidos com PCA, e as quatro colunas seguinte indicam os mesmos parâmetros, porém relativos ao VNS. O número entre parênteses, ao lado do PCA, indica o melhor nível de variância encontrado para a respectiva base de dados. Na última coluna à direita, é indicado o *p-value* (p-valor, ou probabilidade de significância) referente a comparação da acurácia obtida com VNS e com PCA. Bhattacharyya et al. (2011), em seu estudo envolvendo a comparação de Regressão Logística, SVM e *Random Forest*, considerou como diferença significativa um p-valor inferior a 0.01 ($p < 0.01$). Afim de manter uma consistência nos resultados, cada método foi testado vinte vezes em cada uma das bases, de forma que os resultados apresentados a seguir são referentes as médias desses testes.

	PCA (0.8)				VNS				
	Ac.	Prec.	Recall	AUC	Ac.	Prec.	Recall	AUC	p-value (Ac.)
LR	0.8575	0.92	0.81	0.921	0.8746	0.92	0.84	0.923	0.022
kNN	0.8442	0.90	0.82	0.907	0.8725	0.92	0.84	0.917	0.001
Bagging	0.8451	0.87	0.84	0.897	0.8809	0.91	0.87	0.929	< 0.001
Boosting	0.8486	0.87	0.85	0.902	0.8752	0.93	0.84	0.925	0.002
RF	0.8436	0.87	0.84	0.899	0.8873	0.91	0.89	0.928	< 0.001
SVM	0.8497	0.93	0.79	0.920	0.8491	0.93	0.79	0.923	0.937
NN	0.8598	0.92	0.82	0.920	0.8801	0.91	0.87	0.924	0.009

Tabela 10 – Australian Dataset: PCA e VNS

	PCA (0.8)				VNS				
	Ac.	Prec.	Recall	AUC	Ac.	Prec.	Recall	AUC	p-value (Ac.)
LR	0.7374	0.77	0.89	0.774	0.7768	0.80	0.91	0.793	< 0.001
kNN	0.7262	0.75	0.91	0.757	0.7674	0.78	0.93	0.775	< 0.001
Bagging	0.7316	0.77	0.87	0.755	0.7726	0.80	0.89	0.774	< 0.001
Boosting	0.7194	0.77	0.85	0.724	0.7694	0.81	0.88	0.772	< 0.001
RF	0.7316	0.77	0.87	0.757	0.7786	0.80	0.91	0.777	< 0.001
SVM	0.7364	0.77	0.89	0.770	0.7736	0.80	0.89	0.793	< 0.001
NN	0.7382	0.77	0.89	0.774	0.7754	0.80	0.90	0.789	< 0.001

Tabela 11 – German Dataset: PCA e VNS

	PCA (0.9)				VNS				
	Ac.	Prec.	Recall	AUC	Ac.	Prec.	Recall	AUC	p-value (Ac.)
LR	0.7957	0.80	0.98	0.701	0.8056	0.81	0.98	0.708	< 0.001
kNN	0.7912	0.80	0.96	0.670	0.8111	0.83	0.95	0.708	< 0.001
Bagging	0.7964	0.82	0.94	0.703	0.8147	0.84	0.94	0.741	< 0.001
Boosting	0.7977	0.81	0.96	0.697	0.8203	0.84	0.96	0.751	< 0.001
RF	0.8000	0.82	0.94	0.709	0.8174	0.84	0.95	0.749	< 0.001
SVM	0.7776	0.78	1	0.667	0.7776	0.78	1	0.689	1
NN	0.8010	0.83	0.93	0.728	0.8217	0.84	0.95	0.748	< 0.001

Tabela 12 – Taiwan Dataset: PCA e VNS

Com base nos resultados apresentados, não é possível apontar uma combinação de *feature selection* e classificador que seja claramente superior. Entretanto, pode-se observar que alguns métodos tiveram desempenho inferior em determinadas bases de dados, em especial o método SVM na base TC. Nesse caso, a acurácia apresentada, tanto em conjunto com PCA, quanto com VNS, foi de 77,76%, o que a coloca ligeiramente abaixo dos outros métodos, mas observando a medida do *recall*, e levando-se em conta que a base de Taiwan apresenta cerca de 22% de *default*, pode-se inferir que o classificador simplesmente considerou todos os dados como não *default*, independentemente de estar associado com o PCA ou o SVM.

Quando atrelados ao VNS, os classificadores produziram, de maneira consistente, melhores resultados tanto em termos de acurácia, quanto de AUC, mesmo que por vezes por

	PCA (0.9)				VNS				
	Ac.	Prec.	Recall	AUC	Ac.	Prec.	Recall	AUC	p-value (Ac.)
LR	0.8590	0.91	0.82	0.909	0.8659	0.92	0.82	0.915	0.445
kNN	0.8546	0.90	0.83	0.912	0.8769	0.92	0.85	0.917	0.030
Bagging	0.8202	0.83	0.85	0.891	0.8856	0.90	0.89	0.926	< 0.001
Boosting	0.8419	0.86	0.86	0.898	0.8812	0.92	0.86	0.930	< 0.001
RF	0.8465	0.86	0.86	0.903	0.8957	0.92	0.89	0.931	< 0.001
SVM	0.8558	0.93	0.80	0.908	0.8543	0.93	0.80	0.913	0.870
NN	0.8584	0.91	0.83	0.908	0.8725	0.93	0.83	0.918	0.097

Tabela 13 – Japan Dataset: PCA e VNS

	PCA (0.9)				VNS				
	Ac.	Prec.	Recall	AUC	Ac.	Prec.	Recall	AUC	p-value (Ac.)
LR	0.8283	0.83	0.98	0.903	0.8580	0.85	1	0.951	0.001
kNN	0.8306	0.88	0.91	0.868	0.9815	1	0.98	0.989	< 0.001
Bagging	0.8589	0.91	0.91	0.916	0.9880	1	0.98	0.994	< 0.001
Boosting	0.8367	0.89	0.90	0.893	0.9865	1	0.98	0.995	< 0.001
RF	0.8597	0.91	0.92	0.920	0.9871	1	0.98	0.995	< 0.001
SVM	0.8591	0.91	0.91	0.919	0.9750	1	0.98	0.995	< 0.001
NN	0.8553	0.91	0.90	0.920	0.9809	1	0.98	0.992	< 0.001

Tabela 14 – AER Dataset: PCA e VNS

uma pequena margem de diferença. Tal constatação é exemplificada ao se analisar a Tabela 12, relativa a base de dados de Taiwan. Nesse caso, é possível perceber que a diferença média de acurácia entre os classificadores associados ao VNS e PCA ficou entre 1% e 2% (exceto para o SVM, mencionado no parágrafo anterior), ainda sim, nesses casos foi constatado um p-valor significativo. Essa situação pode ser explicada pelo fato de que, apesar de as diferenças médias terem sido moderadas, elas foram constantes ao longo das vinte observações, resultando em uma diferença estatística significativa.

Analisando-se as soluções obtidas ao longo das cinco bases de dados, fica evidente que estas são altamente influentes na geração dos resultados. Nesse sentido, ao contrário do que ocorreu nos demais datasets, os testes aplicados na base AER produziram resultados distintos em relação a utilização do PCA ou VNS, visto que, salvo a Regressão Logística, os demais classificadores apresentaram uma diferença média em termos de acurácia superior a 10% quando associados ao VNS, além de uma AUC também superior. Nessa mesma base de dados, o método *Bagging*, combinado com VNS, registrou a mais alta acurácia do presente estudo, atingindo 98,8% de acerto nas previsões na média das vinte rodadas. Ainda em relação a base AER, é interessante observar que, com apenas duas variáveis em média, o k-NN obteve um ganho expressivo de performance.

Os métodos *ensamble* (*Bagging*, *boosting* e *random forest*) apresentam-se como os mais sensíveis em relação a utilização do PCA ou VNS, visto que foram os únicos a apresentar

diferenças estatisticamente significativas de acurácia em todas as bases de dados. Além disso, em quatro das cinco bases de dados utilizadas, os melhores resultados, em termos de acurácia ou AUC, foram obtidos pela combinação de um método *ensamble* em associação com o VNS, o que indica se tratarem de boas alternativas a serem usadas em conjunto com o seletor de variáveis proposto.

4.2 Resultados sem *feature selection* e com VNS

Com a finalidade de obter mais uma referência comparativa para a análise do método VNS, todos os sete classificadores foram novamente submetidos a vinte rodadas de testes em cada uma das cinco bases de dados. Nas Tabelas 15 a 19, são apresentados os resultados das médias dos sete classificadores, sem a atuação de qualquer técnica de *feature selection*, ou seja, considerando todas as variáveis, e estes são postos frente aos resultados obtidos com o VNS, os quais são os mesmos apresentados nas tabelas anteriores, apenas replicados afim de facilitar a visualização e comparação dos valores obtidos. Seguindo o mesmo padrão adotado anteriormente, a esquerda são indicados os valores de acurácia, *precision*, *recall* e AUC dos classificadores sem *feature selection*, e a direita são mostrados os mesmo parâmetros para os classificadores usados com VNS. Na última coluna à direita tem-se a probabilidade de significância (p-valor) relativa a comparação da acurácia sem seleção de variáveis e com VNS.

	Sem FS				VNS				
	Ac.	Prec.	Recall	AUC	Ac.	Prec.	Recall	AUC	p-value (Ac.)
LR	0.8610	0.91	0.83	0.924	0.8746	0.92	0.84	0.923	0.083
kNN	0.8538	0.91	0.82	0.909	0.8725	0.92	0.84	0.917	0.015
Bagging	0.7604	0.87	0.67	0.903	0.8809	0.91	0.87	0.929	< 0.001
Boosting	0.8529	0.89	0.84	0.924	0.8752	0.93	0.84	0.925	0.011
RF	0.8604	0.88	0.87	0.924	0.8873	0.91	0.89	0.928	0.001
SVM	0.8491	0.93	0.79	0.923	0.8491	0.93	0.79	0.923	1
NN	0.8549	0.88	0.85	0.925	0.8801	0.91	0.87	0.924	0.006

Tabela 15 – Australian Dataset: VNS e sem seleção de variáveis

	Sem FS				VNS				
	Ac.	Prec.	Recall	AUC	Ac.	Prec.	Recall	AUC	p-value (Ac.)
LR	0.7636	0.79	0.89	0.796	0.7768	0.80	0.91	0.793	0.080
kNN	0.7268	0.74	0.94	0.764	0.7674	0.78	0.93	0.775	< 0.001
Bagging	0.7476	0.79	0.87	0.774	0.7726	0.80	0.89	0.774	0.007
Boosting	0.7456	0.79	0.86	0.766	0.7694	0.81	0.88	0.772	0.003
RF	0.7514	0.78	0.89	0.781	0.7786	0.80	0.91	0.777	0.002
SVM	0.7608	0.80	0.87	0.792	0.7736	0.80	0.89	0.793	0.136
NN	0.7548	0.79	0.88	0.792	0.7754	0.80	0.90	0.789	0.003

Tabela 16 – German Dataset: VNS e sem seleção de variáveis

	Sem FS				VNS				
	Ac.	Prec.	Recall	AUC	Ac.	Prec.	Recall	AUC	p-value (Ac.)
LR	0.8022	0.81	0.97	0.714	0.8056	0.81	0.98	0.708	0.048
kNN	0.7915	0.80	0.97	0.678	0.8111	0.83	0.95	0.708	< 0.001
Bagging	0.8117	0.84	0.94	0.741	0.8147	0.84	0.94	0.741	0.023
Boosting	0.8149	0.83	0.96	0.758	0.8203	0.84	0.96	0.751	< 0.001
RF	0.8147	0.84	0.94	0.751	0.8174	0.84	0.95	0.749	0.026
SVM	0.7776	0.78	1	0.695	0.7776	0.78	1	0.689	0.994
NN	0.8168	0.84	0.95	0.748	0.8217	0.84	0.95	0.748	< 0.001

Tabela 17 – Taiwan Dataset: VNS e sem seleção de variáveis

	Sem FS				VNS				
	Ac.	Prec.	Recall	AUC	Ac.	Prec.	Recall	AUC	p-value (Ac.)
LR	0.8552	0.91	0.81	0.916	0.8659	0.92	0.82	0.915	0.259
kNN	0.8549	0.90	0.83	0.912	0.8769	0.92	0.85	0.917	0.028
Bagging	0.8159	0.86	0.79	0.912	0.8856	0.90	0.89	0.926	0.001
Boosting	0.8459	0.89	0.82	0.927	0.8812	0.92	0.86	0.930	0.002
RF	0.8601	0.88	0.87	0.927	0.8957	0.92	0.89	0.931	< 0.001
SVM	0.8543	0.93	0.80	0.912	0.8543	0.93	0.80	0.913	1
NN	0.8416	0.90	0.80	0.917	0.8725	0.93	0.83	0.918	0.004

Tabela 18 – Japan Dataset: VNS e sem seleção de variáveis

	Sem FS				VNS				
	Ac.	Prec.	Recall	AUC	Ac.	Prec.	Recall	AUC	p-value (Ac.)
LR	0.8497	0.84	0.99	0.951	0.8580	0.85	1	0.951	0.249
kNN	0.8470	0.86	0.96	0.869	0.9815	1	0.98	0.989	< 0.001
Bagging	0.9789	0.99	0.98	0.994	0.9880	1	0.98	0.994	< 0.001
Boosting	0.9815	0.99	0.98	0.996	0.9865	1	0.98	0.995	0.003
RF	0.9823	1	0.98	0.995	0.9871	1	0.98	0.995	0.016
SVM	0.9653	0.99	0.97	0.996	0.9750	1	0.98	0.995	0.008
NN	0.9633	0.98	0.97	0.992	0.9809	1	0.98	0.992	< 0.001

Tabela 19 – AER Dataset: VNS e sem seleção de variáveis

Quando comparados os resultados das predições obtidas pelos classificadores sem qualquer técnica de seleção de variáveis e com a presença do VNS, é possível perceber que em quase todas as situações (salvo aquelas utilizando-se SVM) houve um ganho, em termos de acurácia, com a aplicação da seleção de variáveis. Entretanto, na maior parte desses casos, foram constatadas apenas diferenças marginais, e estatisticamente pouco significativas. Entretanto, em determinadas situações, a seleção de variáveis se mostrou altamente relevante para a melhoria dos resultados, em particular no emprego do k-NN, na base de dados AER, em que a acurácia média foi elevada em mais de 13%, e a AUC subiu em 0.12. Também pode-se observar uma considerável evolução na utilização do método *bagging*, na base AC, em que foi registrada

uma acurácia média cerca de 12% superior com o emprego da seleção de variáveis VNS. Esse mesmo método também mostrou uma melhora significativa com o uso do VNS na base JC, com a acurácia média sendo incrementada em quase 7%.

De forma similar ao que aconteceu na comparação entre PCA e VNS, é notório que alguns métodos são mais ou menos sensíveis a aplicação de uma técnica de seleção de variáveis do que outros. Novamente, o método SVM destaca-se como o menos afetado pela utilização de técnicas de *feature selection*. À exceção do AER *Dataset*, nas outras quatro bases de dados o SVM foi o método que apresentou desempenho mais próximo nos testes realizados sem seleção de variáveis e com VNS, sendo que em dois casos, nas bases AC e JC (Tabelas 15 e 18, respectivamente), foi obtido um p-valor de 1, o que indica que não houve nenhum efeito perceptível com relação a acurácia.

Analisando-se apenas o desempenho dos classificadores, sem a realização de nenhuma seleção de variáveis, os métodos *Boosting*, *Random Forest* e Redes Neurais foram os que apresentaram desempenho mais consistente ao longo das cinco bases de dados, visto que, ao contrário dos métodos citados, os outros quatro classificadores apresentaram uma performance consideravelmente inferior aos demais em pelo menos um *dataset*.

De maneira geral, a utilização da seleção de variáveis VNS apresentou, um avanço, mesmo que por vezes discreto, em termos de predição de *default*. Ademais, foram constatados casos em que um classificador originalmente apresentou um fraco desempenho sem a presença de uma técnica de *feature selection*, mas teve seus resultados notavelmente impulsionados com o uso do VNS, o que indica uma alta robustez do método.

A uso do VNS mostrou-se vantajoso em relação ao PCA, uma vez que a acurácia média e AUC foram superiores em praticamente todos os casos analisados. O uso do VNS também se justifica frente a não utilização de qualquer método de *feature selection*, pois seus resultados se mantiveram mais consistentes ao longo dos cinco *datasets* analisados. Em relação aos classificadores, *Boosting*, *Random Forest* e Redes Neurais apresentaram um desempenho constante em todos os testes realizados, ainda sim, a combinação *Random Forest* e VNS liderou o quesito acurácia média em três das cinco bases de dados, sendo a principal indicação de classificador e seletor de variáveis do presente estudo.

5 CONCLUSÃO

Tendo como objetivo principal aumentar a acurácia das predições em modelos de *credit scoring*, por meio da redução da dimensionalidade do espaço de variáveis, foi proposta uma técnica de seleção de variáveis, baseada em um conceito de variação de vizinhanças (VNS). Essa técnica consiste na geração de vários conjuntos de variáveis, e aquele que proporcionar os melhores resultados, é selecionado. Afim de testar a aplicabilidade do VNS, esse foi associado a sete métodos de *machine learning* utilizados para fazer predição acerca da probabilidade de ocorrência de calote (*default*).

Para avaliar a eficácia do método proposto, fez-se a comparação com os resultados gerados a partir da redução de variáveis obtida através do método estatístico PCA, e também com o conjunto total de variáveis, ou seja, sem a utilização de qualquer método de seleção. A principal medida de comparação utilizada foi a acurácia, que mede a proporção entre resultados corretamente previstos sobre o total. Também foram adotadas medidas auxiliares a acurácia, Precision e Recall, além da tradicional medida Area Under the ROC Curve (AUC).

Dos resultados obtidos, pôde-se observar que o VNS teve, de maneira geral, uma melhor performance que o PCA, visto que para maior parte dos casos foi obtida uma diferença significativa em relação à acurácia, além de uma AUC constantemente melhor. Já com relação aos resultados sem nenhuma técnica de feature selection, as diferenças de desempenho foram, em geral, mais sutis, ainda favoráveis ao VNS. Entretanto, alguns casos particulares mostraram que o VNS obteve uma significativa melhora, indicando que se trata de uma técnica robusta.

Não foi apontada nenhuma combinação de seleção de variáveis e classificador claramente superior, entretanto, algumas inferências emergem desse estudo. Entre todos os métodos de *machine learning* utilizados como classificadores, ficou evidente que o SVM é aquele que teve seu desempenho menos afetado por seleção de variáveis. Aliado ao fato de que este teve uma performance particularmente ruim em uma das bases de dados analisadas, SVM torna-se um dos métodos menos interessantes de ser utilizado em conjunto com uma técnica de *feature selection*.

Os métodos *Boosting*, *Random Forest* e Redes Neurais apresentaram resultados satisfatórios, mesmo sem estarem associados a nenhum classificador. Já os métodos k-NN e *Bagging* mostraram resultados ruins em alguns casos. Entretanto, quando atrelados ao VNS, esses cinco métodos apresentam um sólido desempenho em todas as bases de dados, o que indica que o uso de qualquer um destes, em conjunto com o VNS, são, dentre as técnicas analisados, as mais adequadas para serem aplicadas em problemas de credit scoring.

Para estudos futuros, podem ser empregados classificadores não utilizados nesse trabalho. Também seria recomendado adotar uma abordagem híbrida para a etapa de seleção de variáveis, consistindo em um método de pré-seleção associado a uma heurística.

REFERÊNCIAS

ALA'RAJ, M.; ABBOD, M. A systematic credit scoring model based on heterogeneous classifier ensembles. In: IEEE. **2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)**. [S.l.], 2015. p. 1–7.

ALA'RAJ, M.; ABBOD, M. F. A new hybrid ensemble credit scoring model based on classifiers consensus system approach. **Expert Systems with Applications**, Elsevier, v. 64, p. 36–55, 2016.

ANDERSON, R. **The credit scoring toolkit: theory and practice for retail credit risk management and decision automation**. [S.l.]: Oxford University Press, 2007.

ARNER, D. W. The global credit crisis of 2008: Causes and consequences. **Int'l Law.**, HeinOnline, v. 43, p. 91, 2009.

BAESENS, B. et al. Benchmarking state-of-the-art classification algorithms for credit scoring. **Journal of the Operational Research Society**, Taylor & Francis, v. 54, n. 6, p. 627–635, 2003.

BAO, W.; LIANJU, N.; YUE, K. Integration of unsupervised and supervised machine learning algorithms for credit risk assessment. **Expert Systems with Applications**, Elsevier, v. 128, p. 301–315, 2019.

BELLOTTI, T.; CROOK, J. Support vector machines for credit scoring and discovery of significant features. **Expert Systems with Applications**, Elsevier, v. 36, n. 2, p. 3302–3308, 2009.

BEUREN, G. M.; ANZANELLO, M. J. Variable selection using statistical non-parametric tests for classifying production batches into multiple classes. **Chemometrics and Intelligent Laboratory Systems**, Elsevier, v. 193, p. 103830, 2019.

BHATTACHARYYA, S. et al. Data mining for credit card fraud: A comparative study. **Decision Support Systems**, Elsevier, v. 50, n. 3, p. 602–613, 2011.

BREIMAN, L. Bagging predictors. **Machine learning**, Springer, v. 24, n. 2, p. 123–140, 1996.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.

BROWN, I.; MUES, C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. **Expert Systems with Applications**, Elsevier, v. 39, n. 3, p. 3446–3453, 2012.

CHEN, F.-L.; LI, F.-C. Combination of feature selection approaches with svm in credit scoring. **Expert Systems with Applications**, Elsevier, v. 37, n. 7, p. 4902–4909, 2010.

CHOMBOON, K. et al. An empirical study of distance metrics for k-nearest neighbor algorithm. In: **Proceedings of the 3rd international conference on industrial application engineering**. [S.l.: s.n.], 2015. p. 1–6.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine learning**, Springer, v. 20, n. 3, p. 273–297, 1995.

COX, D. R. The regression analysis of binary sequences. **Journal of the Royal Statistical Society: Series B (Methodological)**, Wiley Online Library, v. 20, n. 2, p. 215–232, 1958.

DIETTERICH, T. G. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. **Machine learning**, Springer, v. 40, n. 2, p. 139–157, 2000.

EL-SHERBENY, N. A. Vehicle routing with time windows: An overview of exact, heuristic and metaheuristic methods. **Journal of King Saud University-Science**, Elsevier, v. 22, n. 3, p. 123–131, 2010.

FREUND, Y.; SCHAPIRE, R. E. et al. Experiments with a new boosting algorithm. In: CITESEER. **icml**. [S.l.], 1996. v. 96, p. 148–156.

GELER, Z. et al. Comparison of different weighting schemes for the knn classifier on time-series data. **Knowledge and Information Systems**, Springer, v. 48, n. 2, p. 331–378, 2016.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. [S.l.]: "O'Reilly Media, Inc.", 2017.

GESTEL, T. V.; BAESENS, B. **Credit Risk Management: Basic concepts: Financial risk components, Rating analysis, models, economic and regulatory capital**. [S.l.]: OUP Oxford, 2008.

GOLDBERG, Y.; ELHADAD, M. splitsvm: fast, space-efficient, non-heuristic, polynomial kernel computation for nlp applications. In: **Proceedings of ACL-08: HLT, Short Papers**. [S.l.: s.n.], 2008. p. 237–240.

HAN, L.; HAN, L.; ZHAO, H. Orthogonal support vector machine for credit scoring. **Engineering Applications of Artificial Intelligence**, Elsevier, v. 26, n. 2, p. 848–862, 2013.

HAND, D. J.; HENLEY, W. E. Statistical classification methods in consumer credit scoring: a review. **Journal of the Royal Statistical Society: Series A (Statistics in Society)**, Wiley Online Library, v. 160, n. 3, p. 523–541, 1997.

HENLEY, W.; HAND, D. J. Ak-nearest-neighbour classifier for assessing consumer credit risk. **Journal of the Royal Statistical Society: Series D (The Statistician)**, Wiley Online Library, v. 45, n. 1, p. 77–95, 1996.

HU, L.-Y. et al. The distance function effect on k-nearest neighbor classification for medical datasets. **SpringerPlus**, SpringerOpen, v. 5, n. 1, p. 1304, 2016.

HUANG, J. **Feature selection in credit scoring-a quadratic programming approach solving with bisection method based on Tabu search**. Tese (Doutorado) — Texas A&M International University, 2015.

JOLLIFFE, I. T. Principal components in regression analysis. In: **Principal Component Analysis**. [S.l.]: Springer, 1986. p. 129–155.

KOUTANAEI, F. N.; SAJEDI, H.; KHANBABAEI, M. A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring. **Journal of Retailing and Consumer Services**, Elsevier, v. 27, p. 11–23, 2015.

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

LORENA, A. C. et al. Comparing machine learning classifiers in potential distribution modelling. **Expert Systems with Applications**, Elsevier, v. 38, n. 5, p. 5268–5275, 2011.

MALHOTRA, R.; MALHOTRA, D. K. Evaluating consumer loans using neural networks. **Omega**, Elsevier, v. 31, n. 2, p. 83–96, 2003.

MARQUÉS, A.; GARCÍA, V.; SÁNCHEZ, J. S. Exploring the behaviour of base classifiers in credit scoring ensembles. **Expert Systems with Applications**, Elsevier, v. 39, n. 11, p. 10244–10250, 2012.

MARTENS, D. et al. Comprehensible credit scoring models using rule extraction from support vector machines. **European Journal of Operational Research**, Elsevier, v. 183, n. 3, p. 1466–1476, 2007.

MASSON, E.; WANG, Y.-J. Introduction to computation and learning in artificial neural networks. **European Journal of Operational Research**, Elsevier, v. 47, n. 1, p. 1–28, 1990.

MERTON, R. C. On the pricing of corporate debt: The risk structure of interest rates. **The Journal of finance**, Wiley Online Library, v. 29, n. 2, p. 449–470, 1974.

MLADENOVIĆ, N.; HANSEN, P. Variable neighborhood search. **Computers & Operations Research**, Elsevier, v. 24, n. 11, p. 1097–1100, 1997.

ONG, C.-S.; HUANG, J.-J.; TZENG, G.-H. Building credit scoring models using genetic programming. **Expert Systems with Applications**, Elsevier, v. 29, n. 1, p. 41–47, 2005.

QUINLAN, J. R. Induction of decision trees. **Machine learning**, Springer, v. 1, n. 1, p. 81–106, 1986.

REICHERT, A. K.; CHO, C.-C.; WAGNER, G. M. An examination of the conceptual issues involved in developing credit-scoring models. **Journal of Business & Economic Statistics**, Taylor & Francis Group, v. 1, n. 2, p. 101–114, 1983.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. **Neural networks**, Elsevier, v. 61, p. 85–117, 2015.

SIDDIQI, N. **Credit risk scorecards: developing and implementing intelligent credit scoring**. [S.l.]: John Wiley & Sons, 2012. v. 3.

SIKKA, P. Financial crisis and the silence of the auditors. **Accounting, Organizations and Society**, Elsevier, v. 34, n. 6-7, p. 868–873, 2009.

SONG, F.; GUO, Z.; MEI, D. Feature selection using principal component analysis. In: IEEE. **2010 International Conference on System Science, Engineering Design and Manufacturing Informatization**. [S.l.], 2010. v. 1, p. 27–30.

ŠUŠTERŠIČ, M.; MRAMOR, D.; ZUPAN, J. Consumer credit scoring models with limited data. **Expert Systems with Applications**, Elsevier, v. 36, n. 3, p. 4736–4744, 2009.

TALBI, E.-G. **Metaheuristics: from design to implementation**. [S.l.]: John Wiley & Sons, 2009. v. 74.

THOMAS, L.; CROOK, J.; EDELMAN, D. **Credit Scoring and Its Applications**. [S.l.]: SIAM, 2017. v. 2.

THOMAS, L. C. A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. **International Journal of Forecasting**, Elsevier, v. 16, n. 2, p. 149–172, 2000.

THOMAS, L. C. **Consumer credit models: pricing, profit and portfolios: pricing, profit and portfolios**. [S.l.]: OUP Oxford, 2009.

UĞUZ, H. A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. **Knowledge-Based Systems**, Elsevier, v. 24, n. 7, p. 1024–1032, 2011.

WANG, G. et al. A comparative assessment of ensemble learning for credit scoring. **Expert Systems with Applications**, Elsevier, v. 38, n. 1, p. 223–230, 2011.

WANG, J. et al. Rough set and scatter search metaheuristic based feature selection for credit scoring. **Expert Systems with Applications**, Elsevier, v. 39, n. 6, p. 6123–6128, 2012.

WEST, D. Neural network credit scoring models. **Computers & Operations Research**, Elsevier, v. 27, n. 11-12, p. 1131–1152, 2000.

WEST, D.; DELLANA, S.; QIAN, J. Neural network ensemble strategies for financial decision applications. **Computers & Operations Research**, Elsevier, v. 32, n. 10, p. 2543–2559, 2005.

ZHANG, C. et al. An up-to-date comparison of state-of-the-art classification algorithms. **Expert Systems with Applications**, Elsevier, v. 82, p. 128–150, 2017.

ZHANG, W.; HE, H.; ZHANG, S. A novel multi-stage hybrid model with enhanced multi-population niche genetic algorithm: An application in credit scoring. **Expert Systems with Applications**, Elsevier, v. 121, p. 221–232, 2019.