Proposta de um modelo ensemble para credit scoring

Proposal for an ensemble model for credit scoring

DOI:10.34117/bjdv7n3-232

Recebimento dos originais: 08/02/2021
Aceitação para publicação: 11/03/2021

Tarcísio da Costa Lobato
Mestrado em Estatística
Universidade do Estado do Amazonas - UEA
Endereço: Av Leonardo Malcher, 1141. Escola Superior de Ciências Sociais – Centro, Manaus – AM, Brasil.
E-mail: tlobato@uea.edu.br

Brena do Nascimento Carvalho
Mestrado em Economia Aplicada
Universidade do Estado do Amazonas - UEA
Endereço: Av Leonardo Malcher, 1141. Escola Superior de Ciências Sociais – Centro, Manaus – AM, Brasil.
E-mail: brenanc16@gmail.com

RESUMO
Os modelos de *Credit Scoring* foram desenvolvidos com a finalidade de identificar bons e maus pagadores de financiamentos, de acordo com dados cadastrais que definem seu perfil. Em se tratando de análise de crédito, será utilizado técnicas de aprendizado supervisionado, no qual o objetivo deste estudo envolve a contrução de um modelo estatístico para classificar um cliente entre bom e mau pagador, tomando com base variáveis que descrevem o perfil do mesmo. Os dados de análise de crédito são de difícil acesso, para contornar esse problema foi utilizado a base de dados de acesso livre *German Credit Data*, adquiridos pelo site *UC Irvine Machine Learning Repository.* A metodologia adotada consiste em dividir o conjunto de dados em 80% para treinamento e 20% para teste. Destes 80%, retiram-se 10 amostras aleatórias com reposição de 70% (700 observações). Em cada amostra foi aplicado os classificadores: Naive Bayes, SVM, Regressão Logística, KNN e Árvore de decisão. Desta forma, depois de treinar os classificadores foram aplicados no conjunto de teste, e por sua vez, combinados por votos, gerando a classificação desejada para cada amostra, posteriormente, combinados por voto para gerar a classificação final. A metodologia proposta obteve resultados satisfatórios em comparação com os classificadores mais utilizados na literatura, incluindo métodos de combinação como Bagging, alcançando melhores desempenhos em acurácia e especificidade, e, com um dos menores erros para o falso positivo, ou seja, quando se classifica um cliente como bom sendo que na realidade é mal pagador.

Palavras-chave: Modelo Ensemble, Credit score, Machine Learning, Aprendizado Estatístico.

ABSTRACT
Credit Scoring models were developed with the purpose of identifying good and bad loan payers, according to registration data that define their profile. In the case of credit analysis,

supervised learning techniques will be used, in which the objective of this study involves the construction of a statistical model to classify a customer between good and bad payers, based on variables that describe their profile. Credit analysis data is difficult to access, to circumvent this problem the German Credit Data free access database was used, acquired through the UC Irvine Machine Learning Repository website. The adopted methodology consists of dividing the data set in 80% for training and 20% for testing. Of these 80%, 10 random samples are taken with 70% replacement (700 observations). The classifiers were applied to each sample: Naive Bayes, SVM, Logistic Regression, KNN and Decision Tree. Thus, after training the classifiers, they were applied to the test set, and in turn, combined by votes, generating the desired classification for each sample, subsequently combined by vote to generate the final classification. The proposed methodology obtained satisfactory results in comparison with the classifiers most used in the literature, including combination methods such as Bagging, achieving better performances in accuracy and specificity, and, with one of the smallest errors for the false positive, that is, when classifying a customer as good and in reality it is badly paying.

Keywords: Ensemble Model, Credit score, Machine Learning, Statistical Learning.

1 INTRODUÇÃO

Este artigo pretende desenvolver uma metodologia para um modelo *ensemble* com a finalidade de analisar dados de crédito. Para este propósito, foram utilizadas técnicas de *Machine Learning*.

Como os dados de análise de crédito são de difícil obtenção, optou-se por construir este modelo por meio da base de dados de acesso livre *German Credit Data*, disponibilizada pelo Prof Dr. Hans Hofmann da Universidade de Hamburgo.

Os modelos de *Credit Scoring* foram desenvolvidos com a finalidade de identificar bons e maus pagadores de financiamentos, de acordo com dados cadastrais que definem seu perfil. Para elaboração desses dados, os bancos selecionam as informações cadastrais dos clientes pontuando sua importância em conformidade com suas políticas internas de crédito. Desde modo, é possível se utilizar da estatística e computação para atribuir riscos de crédito aos seus clientes (SANTOS; FAMÁ, 2007).

Para a gestão de crédito é fundamental obter ferramentas que classificquem e auxilie a prever comportamentos de futuras concessões, permitindo uma condução mais eficiente dos recursos, com a vantagem de diminuir a subjetividade no processo e proporcionando maior celeridade as propostas (SOUSA; FIGUEIREDO, 2014).

Será utilizado técnicas de aprendizado estatístico e *Machine Learning* para a proposta do modelo, que por sua vez, é um conjunto de modelos que possuem a finalidade de identificar padrões em um conjunto de dados.

Para Breimann (2001) existem duas culturas no uso de modelos estatísticos: i) *Data modeling culture*: A comunidade estatística que se utiliza dos modelos estatísticos para inferências e previsões; ii) *Algorithmic modeling culture*: A comunidade *Machine Learning* que por meio dos modelos estatísticos, não se preocupa com suas suposições para inferência, apenas se o modelo proposto obtém um bom poder preditivo. Neste caso, será composta por diversos métodos que permitem estimar funções controlando seu risco associado, na tentativa de obter um balanço entre viés e variância do modelo.

Em se tratando de análise de crédito, onde a base de dados será composta por variáveis de entrada (perfil do cliente) e conhecemos a variável de saída (bom ou mau pagador), será utilizado técnicas de aprendizado supervisionado, no qual envolve a contrução de um modelo estatístico para prever ou estimar, tomando com base em variáveis de entra e uma saída do modelo (HASTIE; TIBSHIRANI; FRIEDMAN, 2008).

Com o intuito de verificar a análise de crédito, diversos estudos foram realizados por vários autores utilizando técnicas e modelos híbridos (ABELLÁN; MANTAS, 2014; CUBILES-DE-LA-VEJA *et al.* 2013; KRUPPA *et al.*, 2013; ORESKI; ORESKI, 2014; ZHONG *et al.*2014).

As principais técnicas de aprendizado supervisionado utilizados para contrução do modelo ensemble foram *Naive Bayes*, *Support Vector Machine*, Regressão Logística, KNN, Árvore de decisão (J48). Além desses, utilizou-se para efeito de comparação, os modelos *Bagging* e *Random Forest*.

O presente estudo possui o objetivo de propor uma metodologia para classificar dados de análise de crédito, obtendo um classificador que seja tão bom quanto os já existentes.

2 REFERENCIAL TEÓRICO

2.1 TEORIA DA APRENDIZAGEM ESTATÍSTICA - TAE

A teoria da aprendizagem estatística (TAE) é uma estrutura para o *Machine Learning* a partir dos campos de estatística e análise funcional, tratando o problema de encontrar uma função preditiva baseada em dados (MOHRI *et al.*, 2012).

Os objetivos de aprendizagem são a compreensão e a previsão, sendo dividida em aprendizagem supervisionada, aprendizagem não supervisionada, e aprendizado por reforço. Do ponto de vista da teoria da aprendizagem estatística, a aprendizagem supervisionada é melhor compreendida, pois envolve a aprendizagem de um conjunto de dados de treinamento. Cada ponto no treinamento é um par de entrada-saída, onde a entrada

é mapeada para uma saída. O problema de aprendizagem consiste em inferir a função que mapeia entre a entrada e a saída, de modo que a função aprendida pode ser usada para prever a saída da entrada futura (JAMES *et al*., 2013).

Para se obter bons classificadores a TAE mede sua performance empregando a função risco, obtida pelo risco quadrático, a mesma é decomposta em viés e sua variância. Temos que modelos com muitos parâmetros possuem viés relativamente baixo, porém variância alta, pois é necessária estimá-los todos. Por outro lado, modelos com poucos parâmetros possuem variância baixa, mas víes muito alto, já que são simples para descrever o processo gerador dos dados. Ambos podem ser reduzidos dependendo do classificador adotado, enfatizando a importância de sua escolha adequada (JAMES *et al*., 2013).

Na próxima seção listaremos os principais classificadores para aprendizagem supervisionado, incluindo a descrição de modelos ensemble.

2.2 NAIVE BAYES

O classificador *Naive Bayes* é um dos mais utilizados em *Machine Learning*, devendo-se ao fato da fácil implementação e rapidez na geração dos resultados. É denominado ingênuo por assumir que os atributos são condicionalmente independentes, fato que na prática não pode ser assegurado.

Dado um problema a ser classificado, represente pelo vetor $\boldsymbol{X} = (x_1, \ldots, x_n)$ com n variáveis independentes, atribuindo a esta classe uma probabilidade

$$P(C_k \mid x_1, \ldots, x_n) \tag{1}$$

Usando o teorema de *Bayes*, a probabilidade condicional pode ser decomposta como

$$P(C_k \mid X) = \frac{P(C_k)P(X \mid C_k)}{P(X)} \tag{2}$$

O objetivo é encontrar a classe mais provável dentre todas, utilizando os dados de treinamento $\boldsymbol{X}$, ou seja, a classe com o máximo a posteriori (MAP).

$$C_{MAP} = \arg\max_{C_k \in C} P(C_k \mid X) \tag{3}$$

Substituindo a equação (2) em (3)

$$C_{MAP} = \arg\max_{C_k \in C} \frac{P(C_k)P(X \mid C_k)}{P(X)} \tag{4}$$

Como $P(\boldsymbol{X})$ não depende da classe e será uma constante, pode-se retira-lo sem perda de generalidade.

$$C_{MAP} = \arg\max_{C_k \in C} P(C_k)P(X \mid C_k) \tag{5}$$

Sabemos que este classificador supõe que os atributos sejam condicionalmente independentes dado a classe que pertencem.

$$P(X \mid C_k) = \prod_{i}^{n} P(X = x_i \mid C_k) \tag{6}$$

Fazendo a substituição de (6) em (5) obtemos o classificador *Naive Bayes*.

$$C_{MAP} = \arg\max_{C_k \in C} P(C_k)\prod_{i}^{n} P(X = x_i \mid C_k) \tag{7}$$

Deve-se atentar que a independência condicional é facilmente violada, porém não interfere na boa performance do classificador. No conjunto de treinamento pode ser que não exista algum valor para uma variável qualquer, implicando $P(\boldsymbol{X} = x_i|C_k) = 0$ e $P(C_k|\boldsymbol{X} = x_i) = 0$, nesse caso se usa algum método de correção, como a correção de Laplace.

Se houver alguma variável do vetor X sendo categórica, sua probabilidade é dada por

$$P(X = x_i \mid C_k) = \frac{n^o \text{ classificadas em } C_k \text{ com variável } P(x_1,...,x_n \mid C_k)}{n^o \text{ classificadas em } C} \tag{8}$$

Agora se alguma variável for contínua, uma suposição típica é associar a cada classe distribuídos de acordo com uma distribuição Gaussiana. Primeiro segmentamos os dados pela classe e, em seguida, calculamos a média e variância de x em cada classe.

$$P(x = v \mid c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{-\frac{(v-\mu_c)^2}{2\sigma_c^2}} \tag{9}$$

2.3 REGRESSÃO LOGÍSTICA

Considere agora uma variável aleatória Y, que pode assumir qualquer um dos valores 0 ou 1, $\boldsymbol{X} = (x_1, x_2, x_{n-1})$ um vetor de dimensão $(p\text{-}1)$ de variáveis aleatórias independentes (ou preditoras).

Sejam, também, n observações independentes destas variáveis, escritas na forma, $(Y_i, x_{1i}, x_{2i}, ..., x_{(p-1)i})$, onde $i = 1, 2, ... , n$. O Modelo de Regressão Logística pode também escrito na forma:

$$P(Y_i = 1 \mid X_i) = \frac{e^{\beta_0 + \sum_{j=1}^{p-1} \beta_j X_{ji}}}{1 + e^{\beta_0 + \sum_{j=1}^{p-1} \beta_j X_{ji}}} \tag{10}$$

A equação (10) também é conhecida como Função de Distribuição Acumulada (FDA). O método usado para a estimação dos parâmetros é o mesmo adotado para o caso univariado, Máxima Verossimilhança. As equações de verossimilhança são obtidas pela derivação parcial em relação a cada um dos *p* parâmetros da função $L(\beta)$.

Como no Modelo Logístico Univariado, as equações obtidas com a derivação da Função de Verossimilhança não são lineares, sendo necessários métodos iterativos para a resolução do sistema de equações resultante. O vetor de soluções das equações será representado pela matriz $\hat{\beta}$ e é chamado de Estimador de Máxima Verossimilhança de β. Desta forma, os valores estimados para o Modelo de Regressão Logística Múltiplo são $p^{(Xi)}$, o valor da equação (10) calculada usando a $\hat{\beta}$ matriz e X_i.

Pode-se mostrar que os Estimadores de Máxima Verossimilhança são assintoticamente normais e sua matriz de variâncias e covariâncias, obtida pela matriz de segundas derivadas parciais da função de log-verossimilhança.

2.4 SUPPORT VECTOR MACHINE

Uma técnica que surgiu entre as décadas de 60 e 70, porém as aplicações só começaram a ser realizadas com os avanços computacionais da década de 90.

Os seus resultados são comparáveis com as Redes Neurais, possuindo como principais características, um bom desempenho quando empregado a poucas amostras e mesmo na presença de ruídos, robusta em grandes dimensões e seu ganho é que encontra-se um mínimo globas para a convexidade da função objetivo.

A classificação é baseada na construção de um hiperplano ótimo, obtido por meio da maximização da margem entre os vetores-suporte, segue que o hiplerplano ótimo com margens rígidas é dado por

$$\boldsymbol{w}.\boldsymbol{x} + b = 0 \qquad (11)$$

Onde $\boldsymbol{w}.\boldsymbol{x}$ é o produto escalar entre $\boldsymbol{w}$ que é o vetor normal ao hiperplano e o $\boldsymbol{x}$ de atributos de entrada, b é um termo "compensador" ou bias (LORENA; CARVALHO, 2003).

Seja as seguintes suposições:

$$\begin{cases} wx_i + b \geq +1, \ para \ y_i = +1 \\ wx_i + b \leq -1, \ para \ y_i = -1 \end{cases}$$

Simplificando em:

$$y_i(wx_i + b) \geq 1, \ para \ i = 1, 2, ..., n \qquad (12)$$

Seja *d*+(*d*-) a distância euclidiana entre vetores suporte positivos (negativos) e o hiperplano. A margem será

$$\rho = (d_+ + d_-) \tag{13}$$

A distância de um dado *xi* ao hiperplano será

$$d_i(w,b,x_i) = \frac{y_i(wx_i + b)}{\|w\|} \tag{14}$$

Considerando as suposições da equação (12), temos que

$$d_i(w,b,x_i) \geq \frac{1}{\|w\|} \tag{15}$$

Implicando em:

$$d_+ = d_- = \frac{1}{\|w\|} \Rightarrow \rho = (d_+ + d_-) = \frac{2}{\|w\|} \tag{16}$$

Pela equação (16), nota-se que para maximizar a margem devemos minimizar a norma de ***w***. Então tem-se um problema de otimização não-linear.

Minimizar $\|\boldsymbol{w}\|^2$ sob as restrições da equação (12), pode-se resolver esse problema utilizando multiplicadores de Lagrange.

$$L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i (y_i(wx_i + b) - 1) \tag{17}$$

Fazendo as derivadas parciais iguais a zero, obtém-se os pontos ótimos.

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{18}$$

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i \tag{19}$$

Substituindo as equações (19) e (18) em (17), tem-se um problema de maximização em:

$$\sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j \tag{20}$$

Sujeita a

$$\begin{cases} \alpha_i \geq 0,\ i = 1, 2..., n \\ \sum_{i=1}^{n} \alpha_i y_i = 0 \end{cases} \tag{21}$$

O objetivo é determinar os valores ótimos de (***w***, b), por meio da equação (19), calcula-se ***w**** e b*.

$$w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$$

(22)

O valor de b* é obtido por meio das equações de Karush-Kuhn-Tucker (KKT).

$$w^* = \sum_{i=1}^{n} \alpha_i^* (y_i x_i (w^* x_i + b^*) - 2) = 0,\ i = 1,2,...,n \tag{23}$$

Os vetores-suporte são os $\alpha_i^* > 0$, o hiperplano ótimo é expresso por

$$g(x) = \text{sgn}\left(\sum_{i=1}^{n} \alpha_i^* y_i x_i x^T + b^* \right) = \begin{cases} +1\ se \sum\limits_{x_i \in SV} \alpha_i^* y_i x_i x^T + b^* > 0 \\ -1\ se \sum\limits_{x_i \in SV}^{n} \alpha_i^* y_i x_i x^T + b^* < 0 \end{cases} \tag{24}$$

Pode-se ter também em diversas aplicações a presença de ruídos nos dados, podendo ser não-linear. Nesse caso, tem-se um problema com margens suaves, no qual as variáveis de relaxamento (folga) suavizam as restrições, admitindo a ocorrência de alguns erros de classificação (LORENA; CARVALHO, 2003).

A obtenção do classificador é análoga a com margens rígidas, apenas deve-se considerar as variáveis de fola nos problemas de otimização não-linear.

Existem situações em que o conjunto de dados não são linearmente separáveis, devendo-se empregar métodos não lineares, utilizando uma transformação que leve o conjunto de dados de um espaço de entrada para o espaço de características.

São empregados no vetor **x** as seguintes transformações

Quadro 1 - Sumário dos principais Kernels utilizados nas SVMs.

Tipo de Kernel	Função $K(x_i, x_j)$ correspondente	Comentários
Polinomial	$\left(X_i^T . X_j + 1\right)^p$	A potência *p* deve ser especificada pelo usuário
Gaussiano	$\exp\left(\frac{-1}{2\sigma^2}\left\|X_i . X_j\right\|^2\right)$	A amplitude σ^2 é especificada pelo usuário
Sigmoidal	$\tanh\left(\beta_0 X_i . X_j + \beta_1\right)$	Utilizado somente para alguns valores de β_0 e β_1

Fonte: Haykin (1999).

2.5 K-NEAREST NEIGHBORS (KNN)

Esses classificadores são baseados em memória e não exigem que nenhum modelo se ajuste. Dado um ponto de consulta x_0, encontramos os *k* pontos de treinamento $x(r)$, r = 1,...*K* mais próximo da distância para x_0, e depois se classifica usando o voto maioritário entre os vizinhos *k*.

Por simplicidade, assume-se que as variáveis são de valor real e usamos distância euclidiana no espaço das variáveis:

$$d_{(i)} = \| x_i - x_0 \| \tag{25}$$

Normalmente, padroniza-se cada uma das variáveis para ter média zero e variância 1, uma vez que é possível que sejam medidos em unidades diferentes. Apesar da sua simplicidade, os vizinhos mais próximos de *K* tem sucesso em um grande número de problemas de classificação, muitas vezes, é bem sucedido onde cada classe tem muitos protótipos possíveis, e o limite de decisão é muito irregular.

Tem-se que o ponto de consulta coincide com um dos pontos de treinamentos, de modo que o viés seja zero. Isto é verdade assintoticamente se a dimensão do espaço das variáveis é corrigida e os dados de treinamento preenchem o espaço em um moda densa. Então, o erro da regra *Bayes* é apenas a variância de uma variável aleatória de Bernoulli (variável resposta), enquanto o erro da regra de 1 vizinho mais próximo é o dobro da variância de uma variável aleatória de Bernoulli, uma contribuição cada uma para os alvos de treinamento e consulta.

Partindo de *x*, pegue k^* sendo a classe dominante e $p_k(x)$ a probabilidade condicional verdadeira para a classe *k*. Então

$$erro\ de\ Bayes = 1 - p_{k^*}(x), \tag{26}$$

$$\text{1-nearest-neighbor error} = \sum_{k=1}^{K} p_k(x)(1 - p_k(x)), \tag{27}$$

$$\geq 1 - p_{k^*}(x). \tag{28}$$

A taxa de erro assintótica 1-vizinho mais próximo é a de uma regra aleatória; Escolhe-se tanto a classificação como o ponto de teste aleatoriamente com probabilidades $p_k(x)$, $k = 1,..., K$. Para $K = 2$, pode-se mostrar que a taxa de erro de 1 vizinho mais próximo é:

$$\sum_{k=1}^{K} p_k(x)(1 - p_k(x)) \leq 2(1 - p_{k^*}(x)) - \frac{K}{K-1}(1 - p_{k^*}(x))^2 \tag{29}$$

Este resultado pode fornecer uma idéia aproximada sobre o melhor desempenho que é possível em um determinado problema. Por exemplo, se a regra do 1-vizinho mais próximo tem uma taxa de erro de 10%, então, assintoticamente, a taxa de erro *Bayes* é pelo menos 5%. Para os vizinhos mais simples, o viés e as características de variância podem ditar o número ideal de vizinhos próximos para um determinado problema (KUNCHEVA, 2004).

2.6 ÁRVORE DE DECISÃO

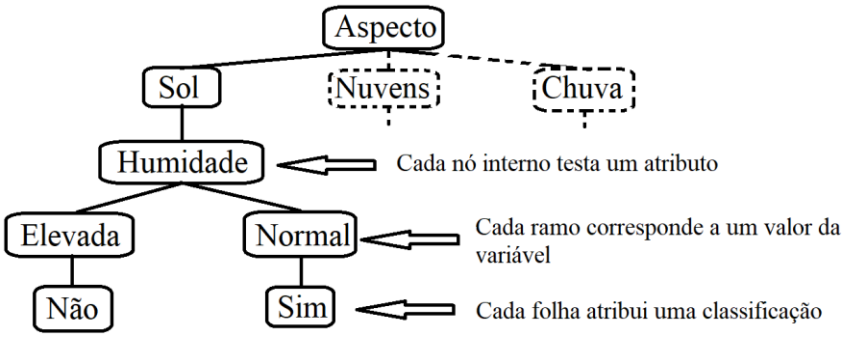
Árvores de decisão são utilizados construindo indução de regras, apresentando resultados hierárquicos. Nesse caso, todos os atributos são apresentados por ordem de importância, com o atributo mais importante no primeiro nó, o segundo mais relevante no segundo nó, assim, sucessivamente.

Um dos primeiros algoritmos desenvolvidos para construção de uma árvore de decisão foi por Quinlan (1993), por meio do algoritmo ID3, agora já é possível encontrar versões mais evoluídas deste, como por exemplo, o algoritmo J48.

A construção de uma árvore de decisão é baseado em predizer classes considerando atributos de um conjunto de dados, por meio da estratégia dividir-para-conquistar, esse é decomposto em subproblemas mais simples, recursivamente, a mesma estratégia é aplicada a cada subproblema. Uma árvore de decisão é eficiente quando os atributos divididos em subespaços, possuem boas característica para associar a uma classe para cada subespaço.

A árvore de decisão é composta por nodos (nós), representando as variáveis (atributos) e de arcos (ramos), provenientes desses nodos e que recebem os valores possíveis para esses atributos. Existem os nodos folha, sendo a associação de cada folha à uma classe, com cada raiz da folha da árvore corresponde a uma regra de classificação (ver Figura 1).

Figura 1: Arquitetura de uma Árvore de Decisão, exemplo clássico das condições meteorológica para decidir jogar tênis.



Fonte: Elaborado pelos autores.

Em uma árvore de classificação, prevemos que cada observação pertence à classe mais comum de observações de treinamento na região a que pertence. Ao interpretar os resultados, muitas vezes estamos interessados não apenas na predição da classe correspondente a uma região de nó terminal particular, mas também nas proporções de classe entre as observações de treinamento que se enquadram nessa região.

Uma vez que planejamos a classificação para atribuir uma observação em uma determinada região à classe de taxas de erro de ocorrência mais comum naquela região, a taxa de erro de classificação é simplesmente a fração das observações de treinamento nessa região que não pertencem aos mais comuns classe:

$$E = 1 - \max_k(\hat{p}_{mk}) \tag{30}$$

Aqui, $\hat{p}_{mk}$ representa a proporção de observações de treinamento na região *M* que são da classe *k*. Contudo, verifica-se que o erro de classificação não é suficientemente sensível para o cultivo de árvores e, na prática, são preferíveis outras duas medidas.

O índice Gini é definido por

$$G = \sum_{k=1}^{K} \hat{p}_{mk}(1 - \hat{p}_{mk}) \tag{31}$$

Uma medida de variância total nas classes *K*. Não é difícil ver que o índice Gini assume um valor pequeno se todas as $\hat{p}_{mk}$ forem próximas de zero ou um.

Uma alternativa ao índice de Gini é a entropia cruzada, dada por

$$D = -\sum_{k=1}^{K} \hat{p}_{mk} \log \hat{p}_{mk} \tag{32}$$

Desde que $0 \leq \hat{p}_{mk} \leq 1$, segue que $0 \leq \hat{p}_{mk} \log(\hat{p}_{mk})$. Pode-se mostrar que a entropia cruzada assumirá um valor próximo de zero se as p_{mk} forem todas próximas a zero ou perto de um. De fato, verifica-se que o índice Gini e a entropia cruzada são bastante semelhantes numericamente.

Ao construir uma árvore de classificação, o índice de Gini ou a cruzamento são tipicamente usados para avaliar a qualidade de uma divisão específica, uma vez que essas duas abordagens são mais sensíveis à pureza do nó do que a taxa de erro de classificação.

2.7 MODELOS ENSEMBLE

Para ser um bom classificador, os algoritmos de aprendizagem supervisionado devem encontrar dentre um espaço de hipóteses a mais adequada para gerar suas previsões, porém nem sempre são capazes de obter um desempenho desejável, pois é uma tarefa árdua.

Diante dessa dificuldade, os modelos Ensembles combinam várias hipóteses com o intuito de encontrar a melhor, gerando classificadores com múltiplas hipóteses com o mesmo conjunto de dados.

O mais indicado no momento da combinação é diversificar os tipos de classificadores, levando a gerar classificações semelhantes para uma hipótese difícil de predizer, ocasionado em uma melhor performance (KUNCHEVA, 2004).

2.7.1 Combinação Por Voto

Cada observação possui uma classificação obtida pelos *B* classificadores para uma determinada classe *C*, portanto a classe mais votada dentre os classificadores será a nova estimativa para a classe (KUNCHEVA, 2004).

2.7.2 Bagging

Esta técnica é baseada na construção de diversos modelos com réplicas *bootstrap* advindas do conjunto de treinamento, combinando as informações dos modelos gerados para encontrar o melhor preditor.

Um modelo *Bagging* terá um bom desempenho na combinação de modelos instáveis, ocasionando variados ajustes para diferentes réplicas *bootstrap*, obtendo mais informações sobre o conjunto de treinamento e produzindo melhores predições. Algoritmos baseados em árvore de decisão provém bons resultados, provenientes de sua instabilidade. (KUNCHEVA, 2004).

2.7.3 Random Forest

Breiman (1996) propõe uma variante do *Bagging* chamado *Random Forest* (floresta aleatória). O *Random Forest* é uma classe geral de métodos de construção de conjunto usando uma árvore de decisão como classificador base. Para ser rotulado como "floresta aleatória", um conjunto de árvores de decisão deve ser construído gerando vetores aleatórios independentemente distribuídos, usando cada vetor para cultivar uma árvore de decisão (KUNCHEVA, 2004).

Assim, *Random Forest* pode ser construído por amostragem a partir do conjunto das variáveis, apenas variando aleatoriamente alguns dos parâmetros da árvore.

2.7.4 Empilhamento

O empilhamento envolve a formação de um algoritmo de aprendizagem para combinar as previsões de vários outros algoritmos de aprendizagem. Primeiro, todos os outros algoritmos são treinados usando os dados disponíveis, então um algoritmo

combinador é treinado para fazer uma previsão final usando todas as previsões dos outros algoritmos como entradas adicionais (BREIMAN, 1996).

Se um algoritmo combinador arbitrário é usado, o empilhamento pode, teoricamente, representar qualquer uma das técnicas de conjunto descritas neste artigo, embora, na prática, um modelo de regressão logística de uma camada seja frequentemente usado como o combinador. O empilhamento geralmente produz melhor desempenho do que qualquer um dos modelos treinados (WOLPERT, 1992).

3 METODOLOGIA

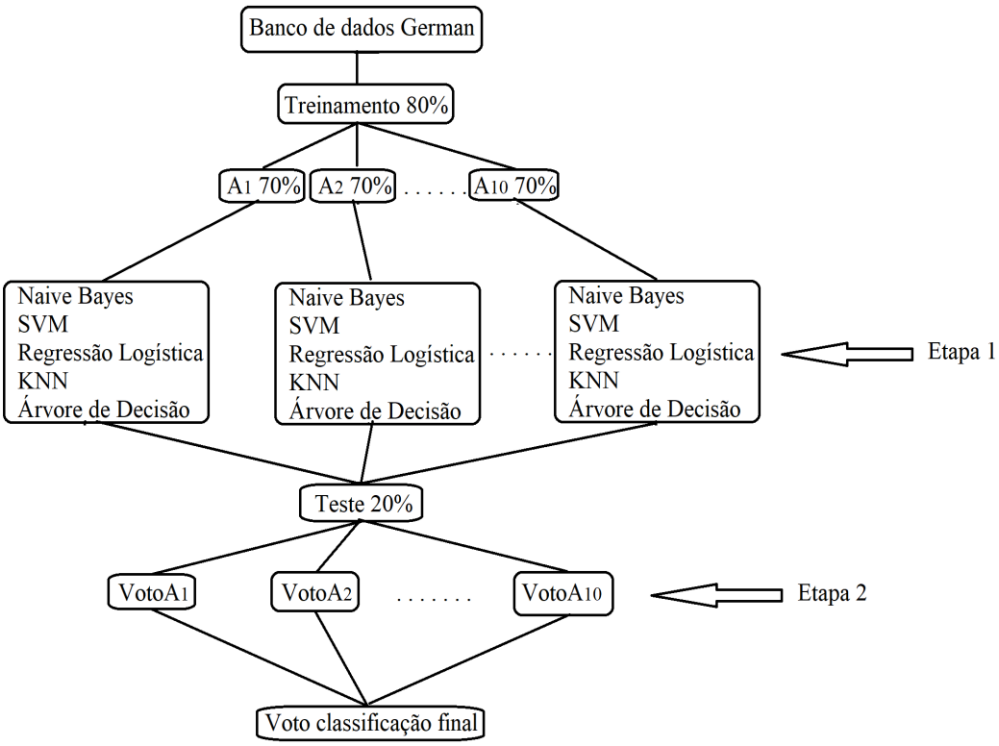
Os dados de análise de crédito são de difícil acesso, para contornar esse problema foi utilizado a base de dados de acesso livre *German Credit Data*, adquiridos pelo site *UC Irvine Machine Learning Repository* e elaborados pelo Prof Dr. Hans Hofmann da Universidade de Hamburgo.

A base de dados é composta por 1000 observações (instâncias), 20 covariáveis ou atributos (7 numéricas e 13 categóricas) e a variável resposta (alvo) sendo a classe dividida em bom e mal pagador. Para análise dos dados foi utilizado o software livre WEKA (*Waikato Environment for Knowledge Analysis*)

Além de verificar as principais medidas de desempenho (Acurácia, Sensitividade, Precisão, Especificidade e F-*measure*) é interessante verificar o menor erro para o Falso Positivo, pois é pior classificar um cliente bom quando eles são ruins, do que é classificar um cliente como ruim quando são bons, ou seja, obter o menor erro para Falso Positivo.

Como pode ser visto na Figura 2, a metodologia adotada consiste em dividir o conjunto de dados em 80% para treinamento e 20% para teste. Destes 80%, retiram-se 10 amostras aleatórias com reposição de 70% (700 observações) (A_1, A_2, …, A_{10}), isso foi realizado para que as 10 amostras possam diferir sem perda de generalidade.

Figura 2: Arquitetura da metodologia empregada.



Fonte: Elaborado pelos autores.

Em cada amostra será aplicado os classificadores: *Naive Bayes*, SVM, Regressão Logística, KNN e Árvore de decisão. Desta forma, depois de treinar os classificadores serão aplicados no conjunto de teste, e por sua vez, combinados por votos, gerando a classificação desejada para cada amostra ($VotoA_1$, $VotoA_2$, …, $VotoA_{10}$), posteriormente, esses serão combinados por voto para gerar a classificação final (Voto classificação final).

Note que existe um empilhamento, pois na Etapa 1 os classificadores são combinados por voto em cada amostra, em seguida, as classificações dessas amostras são combinadas por voto (Etapa 2), portanto a classificação final se utiliza das saídas da primeira etapa como seus dados de entrada, gerando a classificação final.

Para comparar com a metodologia proposta, será aplicado os classificadores já mencionados de forma individual e os modelos ensembles *Bagging* e *Random Forest*, dividindo-se o conjunto de dados em 70% para treinamento e 30% para teste.

Cada classificador foi testado para se verificar qual configuração alcançaria menor erro, essas mesmas configurações foram utilizadas tanto na metodologia quanto para fins de comparação individual.

O classificador SVM obteve melhor desempenho quando utilizado a função de kernel polinomial. O KNN alcançou resultados mais satisfatórios utilizando $k = 3$ vizinhos mais próximos. Na árvore de decisão foi utilizado o algoritmo J48, optou-se por não podá-

la por não trazer ganhos, pois em nosso caso, testando com a poda ao dividir o conjunto de dados gera uma classificação muito diferente do que quando se testa com validação cruzada, sendo muito inconsistente.

O modelo *Bagging* foi treinado e testado utilizando: *Naive Bayes* (Bagging NB), Regressão Logística (*Bagging* RL), SVM (*Bagging* SVM) e KNN (*Bagging* KNN).

4 RESULTADOS E DISCUSSÕES

A Tabela 1 mostra os resultados da porcentagem de acerto da Etapa 1, no qual os classificadores foram treinados nas 10 amostras e aplicados no conjunto de teste, assim como sua combinação por voto.

Tabela 1: Resultados da acurácia dos classificadores e a combinação por voto nas amostras

Amostras	Voto	Naive Bayes	Regressão logística	SVM	KNN 3	Árvore de decisão J48
A1	78.99%	77.31%	78.15%	78.15%	72.27%	68.07%
A2	78.15%	77.31%	78.15%	78.15%	72.27%	68.07%
A3	77.31%	77.31%	77.31%	78.15%	72.27%	68.07%
A4	78.15%	77.31%	77.31%	78.15%	72.27%	68.07%
A5	76.47%	77.31%	76.47%	77.31%	72.27%	68.07%
A6	77.31%	77.31%	76.47%	78.15%	72.27%	68.07%
A7	78.15%	77.31%	77.31%	78.15%	72.27%	68.07%
A8	76.47%	77.31%	76.47%	78.15%	72.27%	68.07%
A9	78.99%	77.31%	78.15%	78.15%	72.27%	68.07%
A10	78.99%	77.31%	78.15%	78.15%	72.27%	68.07%

Pode-se notar que *Naive Bayes*, KNN e J48 tiveram a mesma acurácia nas 10 amostras, exceto em A5 que SVM obteve 77,31%, enquanto a Regressão Logística variou sua acurácia entre as amostras. Verifique que nem sempre a combinação por voto teve o melhor desempenho, porém na maioria dos casos foi igual ou superior.

As médias das correlações entre o desempenho dos classificadores pode ser vista na Tabela 2.

Tabela 2: Matriz das médias das correlações dos classificadores nas 10 amostras.

	Naive Bayes	Regressão logística	SVM	KNN 3	J48
Naive Bayes	1	0.78	0.77	0.48	0.16
Regressão logística		1.00	0.89	0.53	0.32
SVM			1.00	0.58	0.32
KNN 3				1.00	0.20
J48					1.00

Destaca-se a alta correlação ocorrida entre Regressão Logística, *Naive Bayes* e SVM, o que era de se esperar entre *Naive Bayes* e Regressão Logística. As menores correlações são entre KNN e J48.

A comparação entre a porcentagem de acertos do classificador final e os demais pode ser vista na Tabela 3. Nesta, foram selecionados dois classificadores que obtiverem os melhores valores para cada medida de desempenho, exceto para o menor erro do falso positivo que foi selecionado os três melhores.

Tabela 3. Comparação das medidas de desempenho entre os classificadores e a metodologia proposta.

Classificadores	Acurácia	Sensitividade	Precisão	Especificidade	F-mensure	Falso Positivo
Naive Bayes	75.33	84.16	82.67	50.63	83.41	13.00
RL	74.67	84.16	81.94	48.10	8304.00	13.67
SVM	75.67	85.97	81.90	46.84	83.89	14.00
KNN	74.33	84.62	81.30	45.57	82.93	14.33
J48	73.67	86.88	79.34	36.71	82.94	16.67
Bagging NB	74.67	84.16	81.94	48.10	83.04	13.67
Bagging RL	76.67	85.07	83.56	53.16	84.30	12.33
Bagging SVM	77.00	86.43	83.04	50.63	84.70	13.00
Bagging KNN	73.33	84.62	79.57	39.24	80.02	16.00
Random Forest	77.33	90.95	80.72	39.24	85.53	16.00
Voto Final	78.15	87.80	81.82	56.76	84.71	13.45

Nota: Os resultados estão em porcentagem.

Analisando as medidas de desempenho, percebe-se um certo equilíbrio entre o modelo proposto e o *Random Forest*, pois considerando sensitividade e F-*measure* o *Random Forest* obteve melhor desempenho, porém o modelo proposto mostrou-se melhor em acurácia, especificidade e menor erro do falso positivo, sendo essa última muito importante para análise de crédito.

5 CONCLUSÃO

A utilização de métodos de aprendizagem estatístico e *Machine Learning* tem-se mostrado eficiente para *credit score*, tema de grande importância na atualidade.

A metodologia proposta obteve resultados satisfatórios em comparação com os classificadores mais utilizados na literatura, incluindo métodos de combinação como *Bagging*. O modelo proposto alcançou melhores desempenhos em acurácia e especificidade, possuindo um dos menores erros para o falso positivo, ou seja, quando se classifica um cliente como “bom” sendo que na realidade é “mal pagador”.

REFERÊNCIAS

ABELLÁN, J.; MANTAS, C. J. Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring. Expert Systems with Applications, 41(8), 3825–3830. 2014.

BREIMAN, L. Stacked Regression, Machine Learning, 24, 1996.

_____. Statistical modeling: The two cultures. Statistical Science, 16(3):199-231, 2001.

CUBILES-DE-LA-VEGA, M.-D.; BLANCO-OLIVER, A.; PINO-MEJÍAS, R.; LARA-RUBIO, J. Improving the management of microfinance institutions by using credit scoring models based on Statistical Learning techniques. Expert Systems with Applications, 40(17), 6910–6917. 2013.

HASTIE, T.; TIBSHIRANI, R.; FIEDMAN, J. The Elemets of Statistical Learning: Data Mining, Inference, and Prediction. Springer, Second Edition, 2008.

HAYKIN, S. Neural Networks – A Compreensive Foudation. Prentice – Hall, New Jersey, 2 edition. 1999.

JAMES, G.; WITTEN, D.; HASTIE, T.; TIBSHIRANI, R. An Introduction to Statistical Learning with Applications in R. Springer. 2013.

KRUPPA, J.; SCHWARZ, A.; ARMINGER, G.; ZIEGLER, A. (2013). Consumer credit risk: Individual probability estimates using machine learning. Expert Systems with Applications, 40(13), 5125–5131.

KUNCHEVA, L. Measures of diversity in classifier ensembles, Machine Learning, 51, pp. 181-207, 2004.

LORENA, A. C.; CARVALHO, A. C. P. L. F. Introdução às Máquinas de Vetores Suporte Relatórios técnicos do ICMC. São Carlos, 2003.

ORESKI, S.; ORESKI, G. Genetic algorithm-based heuristic for feature selection in credit risk assessment. Expert Systems with Applications, 41(4), 2052–2064. 2014.

QUINLAN, J.C. C4.5: programs for machine learning. San Mateo: Morgan Kaufmann. 302p. 1993.

ROKACH, L. "Ensemble-based classifiers". Artificial Intelligence Review. 33 (1-2): 1–39. 2010.

SANTOS, J. O.; FAMÁ, R. Avaliação da aplicabilidade de um modelo de credit scoring com variáveis sistêmicas e não-sistêmicas em carteiras de crédito bancário rotativo de pessoas físicas. Rev. contab. financ. vol.18, no.44, pág 105-117. 2007.

SOUSA, M. M.; FIGUEIREDO, R. S. Análise De Crédito Por Meio De Mineração De Dados: Aplicação Em Cooperativa De Crédito. Revista de Gestão da Tecnologia e Sistemas de Informação. Vol. 11, No. 2, May/Aug., 2014 pp. 379-396

WOLPERT, D. Stacked Generalization. Neural Networks, 5(2), pp. 241-259. 1992.

ZHONG, H.; MIAO, C.; SHEN, Z.; FENG, Y. Comparing the learning effectiveness of BP, ELM, I-ELM, and SVM for corporate credit ratings. Neurocomputing, 128(27), 285–295. 2014.