INSPER INSTITUTO DE ENSINO E PESQUISA
PROGRAMA DE MESTRADO PROFISSIONAL EM ECONOMIA

ANDREZA LUKOSIUNAS

APLICAÇÃO DE TÉCNICAS DE MACHINE LEARNING EM MODELOS DE ESCORE DE CRÉDITO

São Paulo
2018

ANDREZA LUKOSIUNAS

Aplicação de técnicas de machine learning em modelos de escore de crédito

Dissertação apresentada ao Programa de Mestrado Profissional em Economia do Insper, Instituto de Ensino e Pesquisa, como parte dos requisitos para obtenção do título de Mestre em Economia.

Área de concentração: Economia dos Negócios

Orientador: Prof. Dr. Rinaldo Artes
Coorientador: Prof. Dr. Fábio José Ayres

São Paulo

2018

Lukosiunas, Andreza

Aplicação de técnicas de machine learning em modelos de escore de crédito / Andreza Lukosiunas – São Paulo, 2018

68 f.

Dissertação (Mestrado – Programa de Mestrado Profissional em Economia) – Insper, 2018

Orientador: Prof. Dr. Rinaldo Artes
Coorientador: Prof. Dr. Fábio José Ayres

1. Escore de crédito 2. Aprendizado de máquina 3. Regressão logística 4. Random forests 5. XGBoost 6. Multilayer perceptron

ANDREZA LUKOSIUNAS

APLICAÇÃO DE TÉCNICAS DE MACHINE LEARNING EM MODELOS DE ESCORE DE CRÉDITO

Dissertação apresentada ao Programa de Mestrado Profissional em Economia do Insper, Instituto de Ensino e Pesquisa, como requisito parcial para obtenção do título de Mestre em Economia.

Área de concentração: Economia dos Negócios

Orientador: Prof. Rinaldo Artes
Coorientador: Prof. Fábio José Ayres

Aprovada em:

BANCA EXAMINADORA

_____
Prof. Dr. Rinaldo Artes
Insper

_____
Prof. Dr. Fábio José Ayres
Insper

_____
Profa. Dra. Regina Madalozzo
Insper

_____
Profa. Dra. Airlane Pereira Alencar
IME - USP

Dedico este trabalho a meus pais, Maria Margarida Lukosiunas e Walter Lukosiunas, que estiveram ao meu lado nos bons momentos e carregaram-me nos mais difíceis. A cada dia, tornam-se pais ainda melhores.

AGRADECIMENTOS

Agradeço aos meus pais, que sempre me acolheram, e nas fases mais difíceis do mestrado estiveram ao meu lado.

A minha filha, meu irmão e minha cunhada, que aturaram meu estresse e proporcionaram bons momentos.

Ao Rinaldo, meu orientador, um fofo, que teve muita paciência comigo, me acolhendo da melhor maneira possível, e nesse período foi de suma importância para a expansão dos horizontes do meu conhecimento. Ao Fábio, coorientador, que tinha muito prazer em ensinar-me as técnicas utilizadas neste trabalho.

A Fabi e aos parceiros da turma, sem eles o mestrado teria sido muito mais difícil.

Ao Itaú Unibanco, que colaborou financeiramente com essa brincadeira séria, saio com uma vontade ainda maior de retribuir desenvolvendo um excelente trabalho para esta empresa.

Aos cientistas de dados do Itaú Unibanco, em especial ao Adelmo, que me proporcionaram maravilhosos momentos de aprendizado.

Aos meus gestores Vagner e Felipe por compreenderem que meus 100% foram para 70% quando eu estava fazendo o mestrado, e nem por isso deixaram de acreditar no meu potencial.

A Serasa Experian pela base de dados, o que agregou mais riqueza a este trabalho.

Ao Paulo Dias da Serasa por todas as dicas feras de machine learning.

@s amig@s que entenderam minha ausência nos roles e sempre me deram muita força, principalmente a Danusa, Gabo, Rosie e Cassio.

RESUMO

Visando o aumento do lucro e redução da perda, instituições financeiras credoras esforçam-se em melhorar o acerto ao prever as chances de potenciais devedores ficarem inadimplentes. Com o aumento da capacidade do processamento computacional, técnicas de aprendizado de máquinas estão se popularizando em diversos meios. Diante desses dois cenários, este trabalho propõe a comparação das técnicas regressão logística, *random forests*, *xgboost* e *multilayer perceptron* aplicadas a uma base de escore de crédito disponibilizada pela Serasa Experian contendo o público de pequenas e médias empresas. Foram implementados testes de hipóteses utilizando o teste *DeLong* para comparar as áreas sob a curva roc dos modelos apresentados. A principal contribuição deste trabalho foi mostrar que houve superioridade da técnica random forests quando comparada às outras apresentadas neste trabalho ao diferenciar bons ou maus pagadores.

Palavras chave: escore de crédito; aprendizado de máquina; regressão logística; random forests; gradient boosting; xgboost; multilayer perceptron; redes neurais artificiais; acordo de Basileia.

ABSTRACT

Aiming at increasing profit and reducing loss, creditor financial institutions strive to improve the accuracy by predicting the chances of potential borrowers becoming defaulters. With increasing computational processing capacity, machine learning techniques are becoming very popular in a variety of environments. In the face of these two scenarios, this work proposes the comparison of logistic regression, random forests, xgboost and multilayer perceptron applied to a credit score dataset provided by Serasa Experian containing the public of small and medium enterprises. Hypothesis tests were used with DeLong test to compare the areas under the roc curve of the presented models. The main contribution of this work was to show that there was superiority of the random forests technique when compared to the others presented in this work to differentiate good or bad payers.

Key words: credit scoring; machine learning; logistic regression; random forests; gradient boosting; xgboost; multilayer perceptron; artificial neural network; Basel accord.

SUMÁRIO EXECUTIVO

Modelos de escore de crédito são um conjunto de métodos matemáticos e estatísticos que mensuram o risco de um tomador de crédito entrar em *default*. Esses métodos atribuem uma nota (escore) ao tomador do empréstimo com o objetivo de classificá-lo como possível bom ou mau pagador. A nota é baseada nas idiossincrasias do indivíduo; já a classificação é constituída através de um ponto de corte definido para minimizar a perda média do credor ou aumentar o lucro, em conformidade com o cenário econômico. A partir do escore conferido ao tomador, as instituições credoras, bancos em sua maioria, decidem se vão conceder o crédito e a quantidade que será concedida.

Além da finalidade envolvendo minimizar perda e maximizar lucro sob a perspectiva do credor, os modelos de escore de crédito são de fundamental relevância para o cálculo de alocação mínima de capital, segundo exigências estabelecidas pelo acordo de Basileia, instituídas em consequência dos cenários de crise econômica enfrentados globalmente.

O método mais utilizado atualmente é a regressão logística, sendo crescente o número de estudos na literatura acadêmica aplicando métodos de aprendizado de máquina, que, impulsionados pelo desenvolvimento de métodos de análise para grandes bases de dados (*big data)* e pelo aumento da capacidade de processamento computacional, atraem cada vez mais atenção. Essas técnicas se mostram promissoras ao trazer resultados mais precisos do que a regressão logística.

Neste trabalho, é realizado um estudo aplicando as técnicas regressão logística, *random forests*, *XGBoost* e *multilayer perceptron* para diferenciar o público de pequenas e médias empresas de uma amostra de dados entre bom e mau pagador. O objetivo é avaliar e comparar o desempenho desses métodos na estimação de modelos de escoragem de crédito.

A amostra de dados fornecida pela Serasa Experian, bureau de crédito responsável pela maior e mais completa base de dados cadastrais da América Latina, possui 125.858 observações e 86 variáveis de pequenas e médias empresas de diversas localidades do Brasil e setores da economia. Está dividida em amostra de desenvolvimento, amostra de validação e amostra *out of time* (tradução literal: *fora do*

tempo). Amostra de desenvolvimento é empregada para treinar o modelo; a de validação, para mensurar a qualidade do modelo com o propósito de ajustar seus hiperparâmetros. A *out of time* para averiguar a precisão do modelo para um público em um período diferente das outras amostras.

Mensurar o desempenho do modelo é imprescindível para provar sua superioridade em relação à estimativa aleatória e, também, é uma maneira de constituir um comparativo entre modelos. Neste trabalho, a técnica utilizada para avaliar a performance dos modelos será a área sob a curva roc (auroc, do inglês: *area under the curve roc*).

Escolhida por deter o maior valor, a *auroc* da técnica *random forests* foi comparada com as dos outros métodos. Os testes de hipóteses realizados indicam que a maior *auroc* é a observada na análise dos resultados obtidos pelo método *random forests*.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

Modelos de escore de crédito são um conjunto de métodos matemáticos e estatísticos que mensuram o risco de um tomador de crédito entrar em *default*. Esses métodos atribuem uma nota (escore) ao tomador do empréstimo com o objetivo de classificá-lo como possível bom ou mau pagador. A nota é baseada nas idiossincrasias do indivíduo e reflete a visão da instituição em relação ao cenário econômico futuro (THOMAS *et al.*, 2002); já a classificação é constituída através de um ponto de corte definido para minimizar a perda média do credor ou aumentar o lucro, em conformidade com o cenário econômico. A partir do escore conferido ao tomador, as instituições credoras, bancos em sua maioria, decidem se vão conceder o crédito e a quantidade que será concedida (THOMAS *et al.*, 2002).

Além da finalidade envolvendo minimizar perda e maximizar lucro sob a perspectiva do credor, os modelos de escore de crédito são de fundamental relevância para o cálculo de alocação mínima de capital, segundo exigências estabelecidas pelo acordo de Basileia, instituídas em consequência dos cenários de crise econômica enfrentados globalmente (HUANG; THOMAS, 2015).

Os primeiros problemas para determinar o risco de crédito eram resolvidos pelo julgamento do credor através de decisões de crédito passadas, as primeiras técnicas estatísticas utilizadas nos modelos de escore de crédito foram regressão multivariada e análise discriminante (MYERS; FORGY, 1963). Atualmente, o método mais utilizado é a regressão logística (THOMAS, 2009), sendo crescente o número de estudos na literatura acadêmica aplicando métodos de aprendizado de máquina, muito embora alguns desses métodos permitam pouca ou nenhuma interpretabilidade das variáveis do modelo, essas técnicas se mostram promissoras ao trazer resultados mais precisos do que a regressão logística (LESSMAN *et al.*, 2015). Devido à falta de interpretabilidade, já existe literatura que investigue maneiras de sanar essa condição (RIBEIRO *et al.*, 2016).

Técnicas de aprendizado de máquina aplicadas a modelos de escore de crédito, impulsionadas pelo *big data,* atraem cada vez mais atenção. A explosão de *start-ups* na indústria financeira traz uma mentalidade de que todos os dados podem ser utilizados como dados de crédito, combinando os já estabelecidos com aqueles minerados em atividades offline ou online realizadas por consumidores, os padrões e

sinais dessa vastíssima quantidade de informação podem ser detectados através de algoritmos complexos (HURLEY; ADEBAYO, 2016).

No presente trabalho, será realizado um estudo aplicando as técnicas regressão logística, *random forests*, *XGBoost* e *multilayer perceptron* (pertencente à classe de rede neural artificial) para decifrar se o público de pequenas e médias empresas de uma amostra de dados, disponibilizada pela empresa Serasa Experian, são bons ou maus pagadores. O objetivo é avaliar e comparar o desempenho desses métodos na estimação de modelos de escoragem de crédito.

Os modelos de escore de crédito contribuem para que o crédito, independente da quantia disponível para este fim, seja direcionado para o cliente com menor probabilidade de *default*. O jornal Valor Econômico constatou que em 2018 a inadimplência das empresas diminuiu devido a restrições na oferta de crédito: "*a inadimplência das empresas começou a melhorar de forma mais consistente neste ano, depois de os bancos colocarem o pé no freio na oferta de recursos e constituírem centenas de bilhões de reais em provisões contra perdas. Parte dessas reservas foi desfeita nos últimos meses, sinal de que o risco começou a diminuir*". (Valor Econômico, 2018)

A presente dissertação está organizada em 5 capítulos. A introdução aqui apresentada faz parte do capítulo 1. O capítulo 2 conta com a revisão bibliográfica e teórica de trabalhos que expõem a história dos modelos de escore de crédito e introduzem algumas técnicas de aprendizado de máquina. No capítulo 3 serão apresentadas as metodologias aplicadas para a construção dos modelos. Os resultados encontrados estarão no capítulo 4. E finalmente, no capítulo 5, a conclusão. No apêndice A encontra-se um dicionário de aprendizado de máquina para orientação do leitor.

2. REVISÃO BIBLIOGRÁFICA E TEÓRICA

Neste capítulo, aborda-se a evolução da modelagem de escore de crédito, desde seu surgimento com a análise discriminante até os dias atuais, enfatizando desenvolvimentos recentes.

2.1. Uma breve história em torno de escore de crédito

Anterior à utilização do escore de crédito, a avaliação para concessão de crédito dependia da intuição do credor e de algumas características do tomador, como capacidade de pagamento e garantia atrelada ao empréstimo (THOMAS *et al.*, 2002).

Thomas *et al.* (2002) afirmam que o escore de crédito é uma maneira de distinguir a população em classes. Nessa linha, Fisher (1936) aplicou o princípio da função linear discriminante em um problema taxonômico, onde encontrou a função que melhor discriminava três espécies de flores (*Iris setosa*, *Iris versicolor* e *Iris virginica*) utilizando suas medidas físicas como entradas do modelo. O mesmo princípio de análise estatística foi estudado por Durand (1941) para relacionar características do tomador ao risco, discriminando-os entre bons e maus pagadores, no livro *Risk Elements in Consumer Instalment Financing* destinado a executivos de crédito e estudantes de problemas de crédito (DURAND, 1941).

Com a segunda guerra mundial, Henry Wells construiu um sistema de escore de crédito devido à escassez de analistas de crédito, que eram convocados para o serviço militar. Esse sistema, baseado em técnicas estatísticas possíveis de serem implementadas com baixo poder computacional, permitiu que pessoas com menos experiência pudessem avaliar tomadores em potencial (LEWIS, 1992).

Outros fatores contribuíram para o desenvolvimento do escore de crédito e propiciaram a automação das decisões de empréstimo, o aumento na emissão de cartões de crédito, no fim da década de 60, e o crescimento do poder computacional. Por exemplo, em 1970, empresas como Visa e Mastercard autorizavam bancos a emitirem cartões, que por sua vez passaram a considerar o escore para acelerar no processo de decisão de emissões de cartões (LEWIS, 1992). Em 1980, com o avanço da tecnologia computacional, técnicas como regressão logística e programação linear começaram a ser utilizadas e, mais recentemente, redes neurais (THOMAS *et al.*, 2002).

O objetivo dos credores que utilizam o escore de crédito para concessão de crédito é aumentar o lucro e diminuir as perdas provenientes de maus pagadores. Outra utilidade do escore de crédito é a alocação de capital exigida pelos órgãos reguladores para cada empréstimo; aqueles de melhor qualidade demandam menos capital alocado. Essa medida é oriunda dos acordos de Basileia.

O comitê de Basileia, cuja sede é no *Bank for International Settlement* foi estabelecido em 1974 após distúrbios na moeda internacional e mercado bancário. Sua proposta era melhorar a estabilidade financeira em todo o mundo. O acordo de capital de Basileia (Basileia I), aprovado em 1988, exigia um capital mínimo ponderado pelo risco de crédito dos ativos de 8% e a inclusão de reservas para perdas com empréstimos no cálculo da adequação de capital. Depois disso, foram incorporadas ao documento exigências para risco de mercado oriundo da exposição do banco ao câmbio, títulos de dívida, ações, commodities e opções. Após inúmeras falências de instituições financeiras, em 2004, Basileia I foi substituída por Basileia II, este novo acordo abrangeu três pilares (Bank for International Settlement, 2018):

> “
>
> 1. *Requisitos mínimos de capital, que buscou desenvolver e expandir as regras padronizadas estabelecidas em Basileia I*
> 2. *Revisão supervisora da adequação de capital da instituição e do processo de avaliação interna*
> 3. *Uso eficaz da divulgação como uma alavanca para fortalecer a disciplina de mercado e encorajar práticas bancárias sólidas*
>
> ” – (Bank for International Settlement, 2018).

Com a crise dos *subprimes*, iniciada em 2007, os bancos estavam muito alavancados e com colchões/reservas de liquidez inadequados. Isso veio acompanhado de má governança, gestão de riscos e estrutura de incentivo inadequada, o que foi demonstrado pela má precificação dos riscos de crédito e liquidez e pelo crescimento excessivo do crédito. Com isso, foi desenvolvido o acordo de Basileia III para sanar as deficiências na regulação financeira, que exige um capital mínimo de alta qualidade e colchões/reservas de capital (HUANG; THOMAS, 2015). Em 2010, as propostas de Basileia III foram publicadas:

> “
>
> - *Requisitos mais rigorosos para a qualidade e quantidade do capital regulatório, em especial o reforço do papel central do capital principal*

- *Uma camada adicional de capital principal – colchão/reserva de conservação de capital – que, quando violada, restringe os pagamentos para ajudar a atender à exigência mínima de capital principal*
- *Um colchão/reserva de capital anticíclico, que coloca restrições à participação dos bancos em booms de crédito em todo o sistema, com o objetivo de reduzir suas perdas nos cortes de crédito*
- *Uma taxa de alavancagem – um montante mínimo de capital para absorção de perdas relativas a todos os ativos de um banco e às exposições extrapatrimoniais (fora do balanço), independente da ponderação do risco*
- *Requisitos de liquidez – uma taxa de liquidez mínima (LCR, do inglês: liquidity coverage ratio), destinado a fornecer dinheiro suficiente para cobrir necessidades de financiamento ao longo de um período de 30 dias de estresse; e uma taxa de longo prazo (NSFR, do inglês: net stable funding ratio) destinado a corrigir desfasamentos de prazos ao longo de todo o balanço*
- *Requisitos adicionais para bancos sistemicamente importantes, incluindo absorção adicional de perdas e arranjos reforçados para supervisão e resolução transfronteiriça*

” – (Bank for International Settlement, 2018).

Assim como ocorre a evolução da regulamentação bancária, as técnicas utilizadas para a determinação de escore de crédito também evoluem. Quanto melhor a técnica, menos prejuízo o credor experencia, e menor o impacto negativo que a economia sofre. A próxima seção faz uma breve introdução sobre alguns modelos utilizados em escore de crédito.

2.2. Introdução às técnicas de escore de crédito

Os primeiros modelos de escore de crédito foram desenvolvidos entre 1950 e 1960. Até este período, as principais técnicas estatísticas utilizadas eram análise discriminante e métodos de classificação. Já na década de 1980, técnicas não estatísticas começaram a ser utilizadas (THOMAS *et al.*, 2002): em tais técnicas o modelo de escore de crédito é construído através de um processo de otimização de erro, sem a necessidade de uma formulação probabilística, e validado empiricamente. Atualmente, existe um grande conjunto de técnicas estatísticas e de aprendizado de máquina (do inglês: *machine learning*) utilizado para modelar escore de crédito, muito embora a regressão logística seja ainda a técnica mais comum (THOMAS, 2009). Neste trabalho, as técnicas analisadas serão: regressão logística, *random forest* e *multilayer perceptron*. Avaliaremos também a aplicação da técnica de *ensemble XGBoost*.

2.2.1. Regressão Logística

Em 1974, Delton L. Chesser propôs a utilização de regressão logística em modelos de escore de crédito e, atualmente, este é o modelo mais utilizado para este fim (THOMAS, 2009). Regressão logística é um método estatístico, desenvolvido por David Cox em 1958, para modelar problemas cuja variável resposta fosse binária (sucesso ou fracasso, 0 ou 1). Cox afirma que a probabilidade de um certo resultado binário acontecer depende dos valores das variáveis independentes e propõe a utilização da função logística para o modelo, já que uma relação linear seria inadequada porque as previsões poderiam ultrapassar o intervalo [0, 1] em alguns casos (a curva da regressão logística tem a forma semelhante à da letra S, com assíntotas nos valores zero e um do eixo ordenado). A função logística foi proposta por Pierre-François Verhulst (MINER, 1933).

Mérito de suas vantagens como robustez do modelo, boa interpretabilidade dos resultados, explicação simples e tempo de processamento, a regressão logística é um método bastante utilizado em escore de crédito (Baesens *et al.*, 2003). Alguns pressupostos como variável dependente binária, observações serem independentes uma da outra, baixa ou nenhuma multicolinearidade entre as variáveis independentes são avaliados para validar o modelo.

Esta técnica aplicada para determinar o escore de crédito consiste em prever a probabilidade de um tomador de crédito ser “bom” ou “mau” pagador. O modelo é dado por:

$$p_i = \frac{e^{\boldsymbol{x_i^T \beta}}}{1 + e^{\boldsymbol{x_i^T \beta}}} \tag{3.1}$$

onde p_i é a probabilidade do tomador ser mau pagador, $\boldsymbol{x_i^T} = (1, x_{1i}, x_{2i}, \dots, x_{ki})$ e $\boldsymbol{\beta} = (\beta_0, \beta_1, \dots, \beta_k)^T$.

Após manipular (3.1) e aplicar logaritmo dos dois lados, temos:

$$\ln\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \cdots + \beta_k x_{ki} \tag{3.2}$$

a expressão $\ln\left(\frac{p_i}{1-p_i}\right)$ é denominada logito.

Para estimar os parâmetros, pode-se utilizar o método da máxima verossimilhança, que busca encontrar estimativas que maximizem a função de verossimilhança dada por:

$$L_i = \prod_{y_i=1} p_i \prod_{y_i=0} (1 - p_i) \tag{3.3}$$

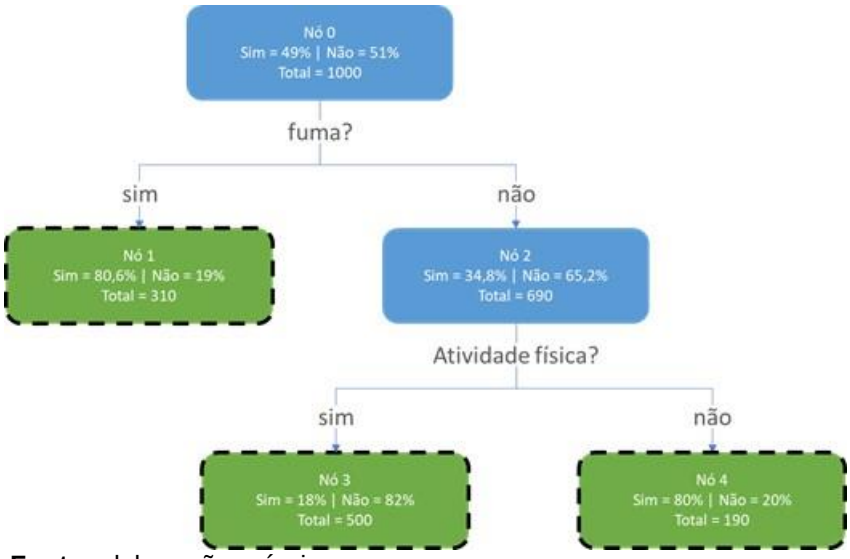
Embora seja possível utilizar o método dos mínimos quadrados (não-linear), recorre-se ao método de máxima verossimilhança por possuir melhores propriedades estatísticas e utiliza-se para ajustar modelos não-lineares (JAMES *et al.*, 2013).

2.2.2. Árvores de Decisão

Árvore de decisão é um procedimento de partição no qual buscam-se as combinações de valores de um conjunto de variáveis independentes que melhor discriminem as respostas dadas a uma variável dependente. O ponto de partida é a identificação, segundo algum critério, da variável independente que melhor discrimina a dependente. A amostra é então particionada utilizando-se o conjunto de valores dessa variável independente, formando grupos que apresentem, segundo algum critério, a maior diferença no comportamento da variável dependente. Esse procedimento é repetido continuamente aos dados de cada partição, até que se atinja um critério de parada. Às combinações de valores das variáveis independentes dá-se o nome de nós.

MEISEL (1973) definiu árvore de decisão como sendo um conjunto de nós, sua estrutura é representada pelo nó que inicia a árvore, chamado de raiz (nó 0 na Figura 1), onde um atributo é escolhido para particionar a amostra em relação à variável resposta, a escolha dessa variável e da partição é estabelecida por intermédio de uma métrica que auxilie na obtenção da partição ótima. As duas divisões da amostra são inseridas em outros dois nós. O algoritmo continua recursivamente até atingir um critério de parada, seja pela escolha do número de níveis ou número mínimo de indivíduos no nó terminal, onde a classe é definida, este nó também é chamado de folha.

Figura 1 - Árvore de decisão



Fonte: elaboração própria

Por exemplo, na Figura 1, temos uma árvore de decisão onde 49% do público teve ataque cardíaco e 51% não teve, com critério de parada quando a folha compreende menos do que 50,1% do público analisado (valores fictícios), as caixas verdes com bordas pontilhadas são as folhas ou nós terminais. A árvore é construída através do particionamento binário da amostra de acordo com as características de indivíduos que sofreram ataque cardíaco (49% da amostra). No primeiro nó (nó 0) ocorre o particionamento da amostra entre fumantes e não fumantes. O nó 1 foi submetido ao critério de parada por conter um percentual de indivíduos da amostra inferior a 50,1%, o que o caracteriza como um nó terminal ou folha. Já no nó 2, ocorreu a partição da amostra entre praticantes de atividade física.

Há diferentes métricas que determinam a variável que será analisada no nó e onde será a partição. No caso de árvore de classificação, como exemplos de métodos utilizados temos: impureza de gini (do inglês: *gini impurity*), ganho de informação (do inglês: *information gain*) e redução de variância (do inglês: *variance reduction*). A classificação é feita através de votação, ou seja, a classe atribuída será a que for maioria na folha. Já na árvore de regressão, é utilizado, em geral, o erro quadrático médio (MSE, do inglês: *mean square error*), e a estimativa é dada pela média das respostas na folha (HEDIBERT, 2017).

2.2.3. Bagging e Boosting

Métodos de *ensemble* são técnicas de aprendizado supervisionado que constroem um conjunto de classificadores para depois combiná-los, de modo que uma nova observação é classificada por meio da votação ou média ponderada das predições de cada um desses classificadores (DIETTERICH, 2000). *Bagging* e *boosting* são métodos de *ensemble*, cujo intuito é aumentar a estabilidade e acurácia dos modelos.

Boosting é uma classe de métodos de *ensemble* destinada a aumentar a acurácia do modelo através da transformação da combinação de preditores fracos (do inglês: *weaklearners*) em um preditor forte (do inglês: *stronglearner*) (BREIMAN, 1996a). *Boosting* foi proposto por Robert E. Schapire em 1990, e consiste em treinar um preditor fraco com elementos da amostra de treino, o próximo preditor é treinado de maneira que o erro do preditor anterior seja corrigido. Alguns dos métodos presentes na classe de *boosting* são: AdaBoost, *gradient boosting* (FRIEDMAN *el al.*, 2001) e *XGBoost* (CHEN; GUESTRIN, 2016).

Breiman (1996) propôs um método para gerar um classificador por meio da combinação de múltiplos preditores, em geral árvores, utilizando *bagging* (do inglês: ***b****ootstrap* ***agg****regat****ing***). Segundo Breiman (1996), o método consiste em extrair amostras com reposição da amostra de treino (*bootstrap*) de tamanhos iguais ao da amostra de treino. Para cada amostra *bootstrap* obtém-se uma árvore de classificação. A predição resultante do processo de *bagging* é determinada pela votação de todos os resultados gerados pela coleção de preditores. O método visa melhorar a acurácia e estabilidade do modelo e diminuir a variabilidade da amostra configurada para treinar o preditor e sobreajuste do modelo. Breiman (1996) conclui que se ganha no aumento da acurácia, mas perde-se na interpretação das estruturas dos preditores.

2.2.4. Random Forest

Random Forests é um método de *ensemble* proposto por Breiman em 2001, ele adicionou ao *bagging* mais um aspecto: a seleção aleatória de uma certa quantidade de variáveis independentes utilizadas em cada partição da árvore; a quantidade de variáveis utilizadas em cada partição é um parâmetro do método. A

finalidade dessa seleção aleatória de variáveis é reduzir o sobreajuste da amostra de desenvolvimento (ou amostra de treino) no modelo e a correlação entre as árvores, que pode acontecer caso uma variável seja fortemente preditiva. A correlação entre as árvores pode ser mensurada através da correlação entre as respectivas respostas. A lei forte dos grandes números garante que adicionar mais árvores ao modelo não o sobreajusta, pois demonstra que o erro converge (BREIMAN, 2001), i.e., o erro diminui até um certo ponto, não chegando a zero. Outra vantagem desse método é a robustez em relação a ruídos[1], Breiman (2001) observou que o aumento na taxa de erro ao injetar 5% de ruído em uma base de dados de câncer de mama era de 43,2% usando um outro método de classificação por árvores e menor do que 12% usando *random forests*. Finalmente, a resposta para cada observação da amostra será a votação (ou média ponderada) da classe estimada por todo o conjunto de árvores.

Matematicamente, Breiman (2001) define *random forests* como:

- Θ_k: k-ésima árvore gerada, com a amostra de desenvolvimento aleatória, independente das árvores $\Theta_1, \dots, \Theta_{k-1}$, mas com a mesma distribuição
- $\boldsymbol{x}$: vetor de variáveis independentes
- $h(\boldsymbol{x}, \Theta_k)$: classificador resultante
- $\{h(\boldsymbol{x}, \Theta_k), \mathrm{k} = 1, \dots\}$: coleção de preditores estruturados em árvores que forma o *random forests*

Na modelagem de escore de crédito, Lessman *et al.* (2015) concluíram que *random forests* obteve melhores resultados do que regressão logística ao prever os resultados de 8 bases de dados de escore de crédito. Brown e Mues (2012), compararam a área sob a curva (AUC, do inglês: *area under the curve*) dos métodos empregados no estudo para prever bons e maus pagadores de 5 bases de dados desbalanceadas de escore de crédito e concluíram que *random forests* performa melhor do que a regressão logística quando aplicado em base de dados de escore de crédito desbalanceada.

2.2.5. Gradient Boosting

Conforme exposto anteriormente, a ideia central do *boosting* é combinar classificadores fracos de modo que se transformem em um classificador forte minimizando o erro a cada passo.

[1] Ruídos são elementos indesejados na amostra que podem ser ocasionados por algum problema na captura ou na transferência dos dados

Friedman (2001) propôs um método baseado no conceito de *boosting* que consiste em obter uma aproximação para a variável resposta que minimize o valor esperado de uma função de perda arbitrária e diferenciável através do *gradient descent*. Esse método foi denominado como *gradient boosting*. O método atua de modo a minimizar o resíduo a cada passo, como elucidado na Tabela 1 (KAGGLE[2]).

Tabela 1 - Processo de aplicação do gradient boosting

Passo	Valor para ajuste	Predição	Predição combinada	Resíduo
1	$\mathbf{y}$	$\hat{\mathbf{y}} = h(\mathbf{y})$	$\hat{\mathbf{y}}_1 = \hat{\mathbf{y}}$	$\mathbf{r}_1 = \hat{\mathbf{y}}_1 - \mathbf{y}$
2	$\mathbf{r}_1$	$\hat{\mathbf{r}}_1 = h(\mathbf{r}_1)$	$\hat{\mathbf{y}}_2 = \hat{\mathbf{y}}_1 + \hat{\mathbf{r}}_1$	$\mathbf{r}_2 = \hat{\mathbf{y}}_2 - \mathbf{y}$
3	$\mathbf{r}_2$	$\hat{\mathbf{r}}_2 = h(\mathbf{r}_2)$	$\hat{\mathbf{y}}_3 = \hat{\mathbf{y}}_1 + \hat{\mathbf{r}}_1 + \hat{\mathbf{r}}_2$	$\mathbf{r}_3 = \hat{\mathbf{y}}_3 - \mathbf{y}$
...				
n	$\mathbf{r}_{n-1}$	$\hat{\mathbf{r}}_{n-1} = h(\mathbf{r}_{n-1})$	$\hat{\mathbf{y}}_n = \hat{\mathbf{y}}_1 + \text{soma}(\hat{\mathbf{r}})$	

Fonte: elaboração própria

Na Tabela 1 foi esquematizada a utilização do *gradient boosting*. Temos:

y: variável dependente

h(.): classificador

y_i: predição no i-ésimo passo

r_i: resíduo no i-ésimo passo

y_n: predição final da variável dependente y

Observe que na coluna **predição** da Tabela 1, do passo 2 em diante o classificador será ajustado ao resíduo, análogo ao ajuste para prever a variável dependente no passo 1.

Entre as funções de perda frequentemente aplicadas (FRIEDMAN, 2001) estão:

1. Mínimos quadrados (do inglês: *least squares*)
2. Menor desvio absoluto (do inglês: *least absolute deviation*)
3. Verossimilhança binomial logística (do inglês: *logistic binomial log-likelihood*)

O *gradient boosting* foi concebido para ser utilizado com outros classificadores, Natekin e Knoll (2013) citam algumas categorias que podem ser empregadas: modelos lineares e árvores de decisão. Friedman (2001) utiliza em seu trabalho

[2] Kaggle é uma comunidade voltada para cientista de dados com materiais de aprendizado de máquina e competições com desafios que fazem uso de métodos de aprendizado de máquina

pequenas árvores de regressão produzidas por CART (do inglês: *classification and regression trees*). Neste trabalho, a estrutura do classificador adotado será em árvore.

Diferente do *random forests*, onde as árvores podem ser construídas paralelamente, no *gradient boosting*, as árvores não são independentes, elas são construídas sequencialmente.

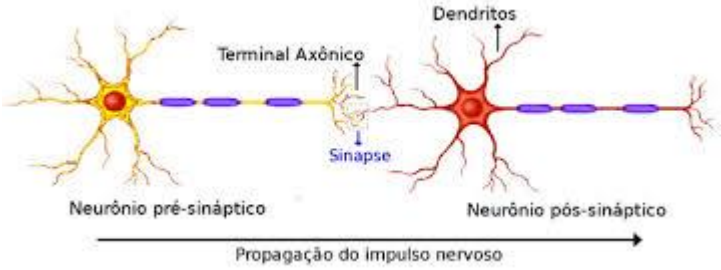
Para obter as predições, foi aplicado o algoritmo *XGBoost - Extreme gradient boosting*, um sistema para implementação de *gradient boosting* teorizado por Chen e Guestrin (2016). Os benefícios trazidos por este sistema, como eficiência, flexibilidade e portabilidade são tão positivos que em 2015, 59% das soluções vencedoras nas competições do Kaggle empregaram o XGBoost (CHEN; GUESTRIN, 2016).

2.2.6. Multilayer Perceptron (Redes Neurais Artificiais)

A utilização de Redes Neurais Artificiais foi iniciada em 1943 com McCulloch e Pitts. Eles propuseram um modelo matemático semelhante à atividade do sistema nervoso neural. Em 1958, Rosenblatt apresentou um modelo, o qual intitulou *perceptron*, para representar como a informação é detectada, armazenada e influencia o reconhecimento e o comportamento do sistema biológico, cuja captação de informação foi inspirado na retina do sistema óptico. *Perceptron* é um classificador binário que combina linearmente os valores de entrada da amostra com pesos, sua saída será 1 se o valor for maior do que o viés e 0, caso contrário. As pesquisas nesse campo foram escassas devido à limitação na capacidade de processamento numérico disponível na época e à inexistência de algoritmos eficientes para o treinamento de redes neurais multicamada, até que a tecnologia permitiu aumento no poder de processamento computacional, o que colaborou para que métodos mais complexos fossem testados.

O processamento e a comunicação de informações no cérebro humano acontecem através de um grande número de dendritos, que carregam sinais elétricos para um neurônio que, por sua vez, transforma-o em pulso de eletricidade que será enviado pelo axônio para as sinapses e, assim, levam a informação para os dendritos de outros neurônios (THOMAS *et al.*, 2002).

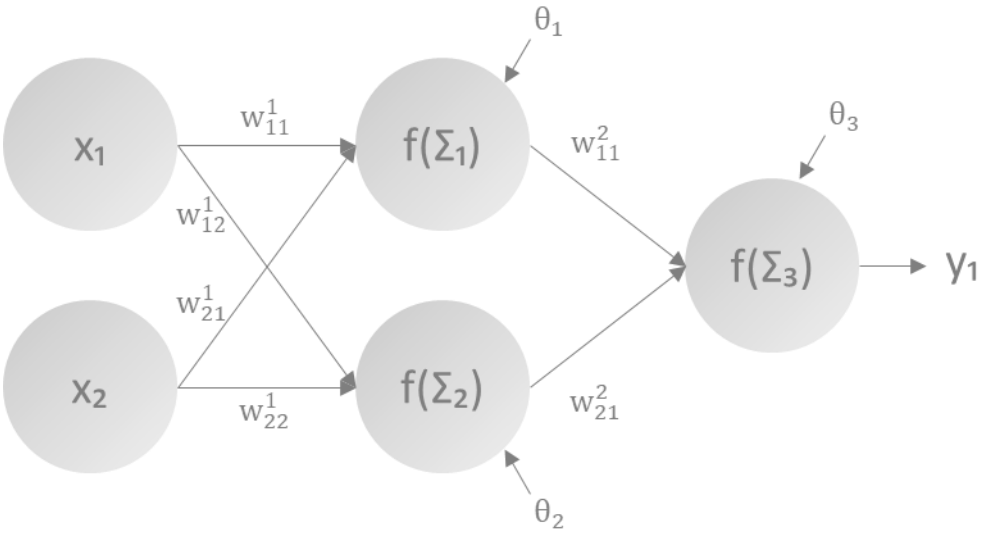
Figura 2 - Comunicação entre neurônios



Fonte: Revista Brasileira de Ensino de Física (2015)

As redes neurais artificiais foram desenvolvidas para tentar simular esse mesmo processamento das redes neurais biológicas, Figura 2. O método está estruturado em camada de entrada (do inglês: *input layer*), camadas escondidas (do inglês: *hidden layers*), camada de saída (do inglês: *output layer*) e as arestas (do inglês: *edge*). As variáveis da base de dados (**X**) formam a camada de entrada, cada aresta possui um peso (**W**) que será aplicado em **X**. Uma função de ativação é aplicada em $\mathbf{X^TW}$ com um viés (do inglês: *bias*) **θ** agregado e, assim, um neurônio da camada escondida é formado, o viés é similar ao intercepto da regressão, fazendo com que a saída seja diferente de zero caso os atributos sejam zero. O processo é repetido para todos os neurônios até atingir a camada de saída.

Figura 3 - Estrutura do multilayer perceptron



Fonte: elaboração própria.

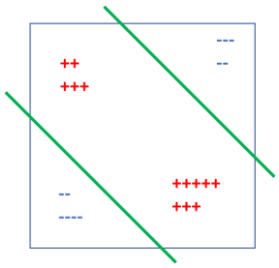
Algebricamente, temos:

$$f(\Sigma_i) = f\left(\sum_{q=1}^{p} w_{qi}x_q - w_i\theta_i\right)$$

onde, $f(\Sigma_i)$ é o i-ésimo neurônio da 1ª camada escondida, w_{qi} é o peso (aresta) aplicado na q-ésima variável, x_q é a q-ésima variável de entrada, w_i é o peso aplicado no i-ésimo viés, θ_i é o i-ésimo viés e é igual a -1 ou 1, e f é a função de ativação.

Utilizar funções de ativação faz com que o aprendizado passe por relações não-lineares. Entre as mais populares estão: degrau, linear, sigmoide logarítmica, tangente hiperbólica, gaussiana e ReLu (do inglês: *rectified linear unit*). Cada neurônio no *hidden layer* representa um hiperplano (linhas verdes) que cortarão o espaço (quadrado azul) como na Figura 4.

Figura 4 - Hiperplanos separando as classes no espaço



Fonte: elaboração própria

Ao atingir a camada de saída, os erros são calculados e distribuídos de volta para a rede, de trás para a frente, ajustando os pesos de modo a reduzir o erro (THOMAS *et al.*, 2002). Esse treino é realizado diversas vezes até que um critério de parada seja atingido, i.e., o número de vezes que haverá um novo treinamento. Uma maneira de realizar esse processo é através do algoritmo denominado como *backpropagation*. *Backpropagation* é um algoritmo que consiste em "*distribuir o erro de volta para a rede proporcionalmente à contribuição feita a ele por cada peso e ajustar os pesos para reduzir essa parte do erro*" – Thomas *et al.*, 2002.

No escore de crédito, o método de redes neurais artificiais alcançou melhores resultados do que regressão logística ao estudarem 8 bases de dados de escore de crédito (LESSMAN *et al.*, 2015). Já Brown e Mues (2012) obtiveram melhores resultados que regressão logística para bases desbalanceadas que continham menos do que 5% de maus pagadores.

2.2.7. Outros modelos de escore de crédito na literatura

Abellán e Castellano (2017) expuseram que modelos de estatística clássica assumem conhecimentos prévios que não são necessários quando ferramentas de inteligência artificial são aplicadas.

Lessman *et al.* (2015) compararam 41 algoritmos de classificação em 8 base de dados de escore de crédito. Entre regressão logística, *random forests* e rede neural artificial, concluíram que a predição da regressão logística é a menos acurada, e que não há evidências suficientes para dizer que *random forests* e rede neural artificial tiveram desempenho significativamente diferentes.

Wang *et al.* (2011) testaram os métodos de *ensemble bagging* e *boosting* nos classificadores: análises de regressão logística (LRA, do inglês: *logistic regression analyses*), árvore de decisão (DT, do inglês: *decision tree*), rede neural artificial (ANN, do inglês: *artificial neural network*) e máquina de vetores de suporte (SVM, do inglês: *support vector machine*) em três bases de dados para classificação de crédito. Citam que, para um método de *ensemble* ser bom, ele deve satisfazer a acurácia e, citam ainda que, melhorar 1% da acurácia ao identificar maus pagadores diminui bastante a perda financeira da instituição. Após o estudo, concluíram que os métodos de *ensemble* trazem resultados significativamente melhores do que a utilização dos classificadores individualmente. O método de *ensemble bagging* performou melhor do que o *boosting*. Freund e Schapire (1997) sugerem que isso pode acontecer devido ao sobreajuste (do inglês: *overfitting*) na base de treino, pois as observações classificadas erroneamente têm mais peso e estas podem tratar-se de ruídos, portanto *bagging* é preferível quando o ruído não é removido da amostra de treino.

E, ainda, Wang *et al.* (2011) consideram *bagging* aplicado à árvore de decisão um dos melhores modelos. Já a aplicação de métodos de *ensemble* em SVM trouxe piores resultado no erro do tipo I, que pode ser devido ao sobreajuste na base de dados de treino para observações de bons pagadores. Pontuaram como limitação dos métodos de *ensemble* a falta de interpretabilidade dos resultados.

Opdal *et al.* (2017) mostram que técnicas de aprendizado de máquina trazem mais lucro do que regressão logística, em especial quando seus hiperparâmetros são ajustados. Na rede neural, utilizar menos camadas fez com que o modelo resultasse em melhor acurácia, pois mais camadas podem sobreajustar o modelo.

3. METODOLOGIA

Neste trabalho, foram implementados 4 métodos para prever se um cliente será bom ou mau pagador a partir de características disponíveis em uma base de dados de escore de crédito. Os métodos estudados são regressão logística, *random forests*, *xgboost* e *multilayer perceptron*. Cada um deles possui particularidades e requer diferentes ajustes, que serão detalhados.

3.1. Base de dados

Para avaliar os métodos utilizados neste trabalho, uma amostra de dados foi fornecida pela Serasa Experian, bureau de crédito responsável pela maior e mais completa base de dados cadastrais da América Latina. A amostra possui 125.858 observações e 86 variáveis de pequenas e médias empresas de diversas localidades do Brasil e setores da economia, com mais de 1 ano de fundação e histórico de restrição, i.e., quando os credores não emprestam crédito por algum motivo. A descrição das variáveis da amostra está no apêndice B.

A amostra já contemplava a divisão em desenvolvimento (DES), validação (VAL) e *out of time* (OOT, tradução literal: *fora do tempo*), suas quantidades podem ser vistas na Tabela 2. As amostras de desenvolvimento e validação compreendem o período de janeiro a dezembro de 2015, já *a out of time*, de janeiro a junho de 2016. Amostra de desenvolvimento é empregada para treinar o modelo; a de validação, para mensurar a qualidade do modelo com o propósito de ajustar seus hiperparâmetros. Existe também a amostra de teste para avaliar a qualidade do modelo após todos os ajustes. Neste trabalho, a amostra *out of time*, que averigua a precisão do modelo para um público em um período diferente das outras amostras, será análoga à amostra de teste.

Tabela 2 - Números de bons e maus pagadores

Amostra	Quantidade	Proporção
Desenvolvimento		
Bons	50.869	91,6%
Maus	4.663	8,4%
Total	**55.532**	**44,1%**
Validação		
Bons	22.039	91,6%
Maus	2.029	8,4%
Total	**24.068**	**19,1%**
Out of time		
Bons	42.103	91,0%
Maus	4.155	9,0%
Total	**46.258**	**36,8%**
Total da base		
Bons	115.011	91,4%
Maus	10.847	8,6%
Total	**125.858**	**100%**

Fonte: elaboração própria

Na base de dados, a variável DT_T0, informa quando os dados de cada empresa foram captados, esses dados são utilizados como variáveis entrada, ou variáveis independentes, do modelo. A Figura 5, linha do tempo da janela de observação das características preditoras do cliente e apontamento de restritivo, mostra que os dados são referentes a até 5 anos anteriores à data DT_T0 de captura. A variável de saída, ou variável dependente, intitulada como TARGET exprime se nos 12 meses após DT_T0 o cliente obteve atribuição de mau pagador incluída por alguma instituição financeira ou empresa de outro segmento. Essa atribuição é concedida em consequência de determinados tipos, quantidade e valores de restritivos como, por exemplo, atrasar o pagamento de uma dívida, entrada em processo de falência pela empresa, suspeita de lavagem de dinheiro, impedimento pelo poder público. O conceito utilizado para conferir o valor 1 na variável TARGET, que expressa se o cliente é mau pagador, é o *ever*, i.e., se o cliente regularizar a dívida nos 12 meses subsequentes a DT_T0, ao invés de ser marcado como bom pagador, ele permanecerá como mau pagador.

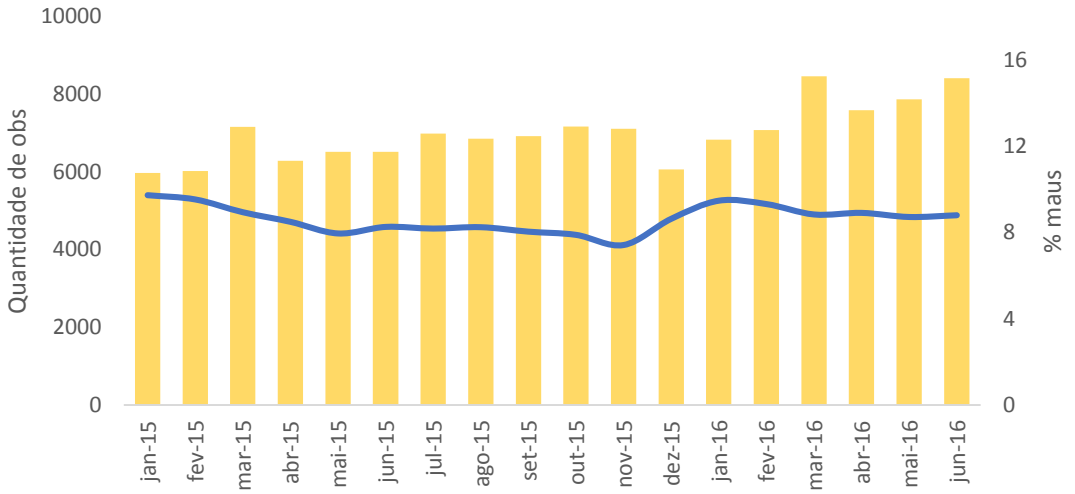
Figura 5 - Linha do tempo

Fonte: elaboração própria

Estão contidas na base de dados variáveis quantitativas discretas que concernem a consultas, restrições e cadastro como, por exemplo, idade de fundação da empresa, tempo desde a entrada do último sócio, quantidade de dívidas vencidas negativadas ou regularizadas, quantidade de protestos vencidos, quantidade de cheques sem fundos vencidos negativados ou regularizados, período de tempo sem dívida, quantidade de consultas, tempo médio entre consultas, período máximo e médio que ficou sem dívida negativada, quantidade de ações vencidas negativadas e quantidade de dívidas vencidas regularizadas, compreendendo um período de tempo de 30 dias a 5 anos. Os dados descritos podem ser referentes à própria empresa ou aos sócios pessoa física (PF) ou pessoa jurídica (PJ) da empresa em questão.

Os dados observados são retratados ao longo do tempo, um gráfico foi construído para que a estabilidade no tempo da quantidade de observações e percentual de maus pagadores fosse analisada. Examinando a Figura 6, nota-se que houve leve aumento no número de observações no tempo, cuja média está em torno de 7.000 observações por mês, já no percentual de maus pagadores (média = 8,6% por mês) verifica-se estabilidade.

Figura 6 - Quantidade de observações e percentual de maus pagadores no tempo



Fonte: elaboração própria

A base original é desbalanceada entre a proporção de bons e maus pagadores. No presente trabalho, os testes serão feitos na base desbalanceada. Os números de bons e maus pagadores referentes a cada amostra são apresentados na Tabela 2, bem como a proporção entre cada uma das classes. Em média, a proporção de maus pagadores é de 8,6%, enquanto de bons pagadores é de 91,4%.

Preparar e tratar a base de dados (do inglês: *dataprep*) antes de aplicar a técnica é de fundamental importância para obter melhor performance do modelo. Por isso, foram retiradas da base variáveis que continham observações faltantes (do inglês: *missing*), com isso o número de variáveis caiu para 67 no total.

Todos os métodos estudados neste trabalho foram desenvolvidos em conformidade com a base de dados cujas variáveis foram categorizadas, o pacote do R utilizado para esse fim foi o *smbinning* (Package R *smbinning*)[3].

Ao final do processo, seis variáveis foram consideradas não informativas, pelo método do valor de informação, IV (SIDDIQI, 2012), fazendo com que o número total de variáveis independentes caísse para 61[4].

[3] Apêndice D
[4] Apêndice C

A métrica de mensuração de performance dos métodos de estimação aplicado aos dados é área sob a curva ROC (AUROC, do inglês: *area under the curve roc*). Esse indicador de performance suporta classes desbalanceadas (FAWCETT, 2006)[5].

Um teste de hipóteses será aplicado para provar se há superioridade de algum modelo entre os melhores de cada técnica expostos neste trabalho, o método utilizado será o teste DeLong (DELONG, 1988). O teste de hipóteses proposto será avaliado com 5% de significância:

H_0: As áreas sob as curvas ROC dos modelos são iguais.
H_1: A diferença entre as áreas das duas curvas é diferente de zero.

3.2. Regressão logística

Foi ajustado um modelo com as 61 variáveis selecionadas para a análise. A macro utilizada para implementação foi a *glm* do pacote do R *smbinning.*

3.3. Random Forests

Random Forests pode ser ajustado por meio de alguns hiperparâmetros. Foram especificados:

- Quantidade de árvores: {1; 100; 500}
- Quantidade de variáveis a serem testadas na partição: {1; 8; 15}
- Quantidade mínima de observações nas folhas: {50; 500; 1000}

Uma variável será escolhida aleatoriamente, entre a quantidade de variáveis adotada nos hiperparâmetros, para que se realize uma partição em um dos nós da árvore. No caso de árvore de classificação, para seleção da melhor partição de uma variável dentro do conjunto que define cada nó, pode ser utilizado o índice Gini, que mede a impureza do nó em relação a variável resposta (ou variável dependente). A Impureza Gini de um dos lados do nó 0 é calculada da seguinte maneira:

$$IG_{lado} = 1 - [prob(\text{nº } resposta\ no\ lado = 1)]^2 - [prob(\text{nº } resposta\ no\ lado = 0)]^2$$

onde lado = {esquerdo, direito}.

[5] Apêndice E

Em seguida, a Impureza Gini Total do nó 0 é calculada:

$$IGT_{\text{nó }0} = \left(\frac{n^{\circ}\ obs_{esquerdo}}{n^{\circ}\ obs_{total}}\right) IG_{esquerdo} + \left(\frac{n^{\circ}\ obs_{direito}}{n^{\circ}\ obs_{total}}\right) IG_{direito}$$

A variável com a partição que tiver o menor IGT no nó 0 será a escolhida para particionar a amostra em outros dois nós. O mesmo processo será utilizado recursivamente para selecionar as variáveis nos nós subsequentes, estes são chamados de nós internos. Os nós terminais são denominados como folhas e são atingidos obedecendo algum critério de parada e, também, é um hiperparâmetro do modelo.

No caso de árvore de regressão (HEDIBERT, 2017), modo empregado neste trabalho por tratar-se de uma amostra desbalanceada contendo apenas duas classes, a seleção de variáveis e a partição são feitas através da minimização do erro quadrático médio (MSE, do inglês: *mean square error*):

$$(\hat{j}, \hat{t}) = \underset{(j,t)}{arg\ min} \left(\sum_{i:x_i \epsilon R_j} \left(y_i - \hat{y}_{R_j}\right)^2 + \sum_{i:x_i \epsilon R_t} \left(y_i - \hat{y}_{R_t}\right)^2 \right)$$

onde, $\hat{y}_{R_j}$ e $\hat{y}_{R_t}$ são as médias das respostas dos dados de treinamento que pertencem às regiões R_j e R_t provenientes do mesmo nó, i é o número da observação, y_i é a resposta observada de i e x_i é o valor da variável independente de i.

Neste trabalho, foi utilizado o pacote do R *randomForest* (LIAW; WIENER, 2002).

3.4. XGBoost

O método *XGBoost* foi implementado com o pacote do R *xgboost* (CHEN; HE, 2018), inclui algoritmo de aprendizagem em árvore, faz computação paralela automaticamente, sendo cerca de 10 vezes mais rápido do que *gradiente boosting machine* (CHEN; HE, 2018). Para a construção dos modelos foram utilizados os seguintes hiperparâmetros:

- Booster: gbtree (classificadores com estrutura de árvores)
- Profundidade máxima das árvores (max_depth): {1; 2; 3}

- Taxa de aprendizado (eta): {0,3; 0,5; 0,7}
- Número de árvores (nrounds): {10; 15; 20}

Subamostras de observações e de variáveis foram incluídas no teste, assim como é feito no *Random Fosrests*, porém, não trouxeram melhora na acurácia, portanto, utilizou-se a configuração padrão para ambas, que é de 100% da amostra. Hiperparâmetros relativos à eficiência computacional foram configurados como padrão.

3.5. Multilayer perceptron

Proveniente da classe de redes neurais artificiais, o *multilayer perceptron* possui diversos hiperparâmetros para ajuste, neste trabalho utilizaremos o pacote *RSNNS* do R (BERGMEIR; BENITEZ, 2012) e focaremos em:

- Número de camadas: 2
- Número de neurônios em cada camada: {10; 30; 50; 100}
- Função ativação: Act_Logistic
- Número de iterações: {1; 5; 10}
- Método de aprendizado: Backpropagation
- Taxa de aprendizado: 0,2
- Erro máximo: 0,001

4. RESULTADOS

Nesta seção são apresentados os resultados da aplicação dos métodos à base de dados. Modelos com área sobre a curva roc maior são considerados melhores, pois o tamanho da área mostra a capacidade de classificação do modelo.

4.1. Regressão logística

A Tabela 3 traz os resultados do Modelo 1:

Tabela 3 - Nº de vars e auroc da regressão logística

Mod	nº vars indep	Auroc des	Auroc val	Auroc oot
1	61	0,838	0,836	0,833

Fonte: elaboração própria

A *auroc* da amostra *out of time* do modelo, que será comparada com as dos outros métodos, é igual a 0,833. É interessante notar que o valor da *auroc* da amostra de desenvolvimento é muito próximo do valor da *auroc* da amostra *out of time*, o que pode indicar que não houve sobreajuste do modelo. Há sobreajuste quando a *auroc* da amostra de desenvolvimento for muito maior do que a *auroc* da amostra de validação ou *out of time*.

4.2. Random Forests

Foram comparados 27 modelos com a técnica *random forests*, os hiperparâmetros utilizados estão apresentados na Tabela 4, e os modelos nela expressos foram ordenados pela *auroc* dos dados *out of time*. O modelo 1 possui auroc para a amostra *out of time* igual a 0,836, a maior quando comparada às dos outros modelos propostos.

Tabela 4 - Hiperparâmetros e auroc do *random forests*

Mod	nº árvores	nº variáveis	nº mín de obs nas folhas	Auroc des	Auroc val	Auroc oot
1	500	8	50	0,958	0,841	0,836
2	500	15	500	0,877	0,840	0,835
3	100	8	50	0,957	0,838	0,835
4	500	8	500	0,872	0,839	0,835
5	100	15	500	0,876	0,839	0,835
6	100	8	500	0,871	0,839	0,834
7	500	15	50	0,963	0,839	0,834
8	500	15	1000	0,858	0,837	0,833
9	100	15	50	0,963	0,837	0,833
10	100	15	1000	0,858	0,837	0,833
11	500	8	1000	0,855	0,837	0,833
12	100	8	1000	0,855	0,837	0,832
13	500	1	50	0,879	0,835	0,830
14	100	1	50	0,879	0,835	0,830
15	500	1	500	0,844	0,832	0,827
16	100	1	500	0,842	0,831	0,826
17	500	1	1000	0,838	0,831	0,825
18	100	1	1000	0,836	0,829	0,825
19	1	15	1000	0,828	0,817	0,807
20	1	8	1000	0,827	0,809	0,805
21	1	8	500	0,836	0,807	0,800
22	1	15	500	0,838	0,806	0,797
23	1	15	50	0,868	0,742	0,735
24	1	8	50	0,862	0,739	0,735
25	1	1	50	0,569	0,550	0,542
26	1	1	500	0,568	0,555	0,542
27	1	1	1000	0,553	0,542	0,536

Fonte: elaboração própria

Portanto, o Modelo 1 será comparado aos melhores modelos das outras técnicas. A partir do valor da *auroc* da amostra de desenvolvimento do Modelo 1 pode-se concluir que se chegou próximo a um caso de sobreajuste, pois a *auroc* da amostra de desenvolvimento está próxima de 100% e é 10% maior do que a *auroc* das outras. No entanto não houve prejuízo no valor da *auroc* da amostra *out of time*.

É interessante notar pela Tabela 4 que ao utilizar apenas 1 árvore no modelo, a *auroc* caia significativamente, e que selecionar 15 variáveis para testar em cada partição pode não aumentar a *auroc* como desejado, e ainda aumenta bastante o tempo de processamento.

Breiman (2001) sugere que a quantidade de variáveis utilizadas em cada partição (nó) seja próxima de $log_2(M+1)$, onde M é a quantidade total de variáveis. Muito embora Breiman tenha utilizado modelos com apenas uma variável, e ficou surpreso com os bons resultados encontrados ao utilizar esse tipo de *random forests*, denominado como *Forest-RI*. Por exemplo, aplicando o *Forest-RI* a uma amostra de dados de câncer de mama, o percentual de erro foi de 2,7% contra 2,9% do *random forests* com mais variáveis, além disso, o tempo de processamento computacional é menor. A documentação do R sugere que um número de variáveis para classificação é de $\sqrt{M}$, e para regressão é $M/3$.

4.3. XGBoost

Para fazer a análise de performance utilizando a técnica *XGBoost*, foram propostos 27 modelos com as configurações que estão Tabela 5, entre os hiperparâmetros estão a profundidade máxima da árvore, a taxa de aprendizagem e o número de árvores. Ordenou-se pelo valor da *auroc* da amostra *out of time*, a maior delas foi a do modelo 1 com 0,833, que será a escolhida para comparação com as dos modelos dos outros métodos. Observando o valor da auroc da amostra de desenvolvimento, concluímos que não houve sobreajuste.

Tabela 5 - Hiperparâmetros e auroc do *XGBoost*

Mod	Prof. Máx.	Taxa de aprendizagem	nº de árvores	Auroc des	Auroc val	Auroc oot
1	3	0,3	20	0,844	0,838	0,833
2	3	0,5	20	0,850	0,841	0,833
3	3	0,5	15	0,846	0,839	0,832
4	3	0,7	15	0,848	0,838	0,832
5	3	0,7	20	0,851	0,837	0,831
6	2	0,5	20	0,841	0,835	0,831
7	3	0,7	10	0,842	0,835	0,831
8	3	0,3	15	0,840	0,835	0,830
9	2	0,3	20	0,837	0,833	0,830
10	2	0,5	15	0,838	0,834	0,830
11	3	0,5	10	0,839	0,834	0,829
12	2	0,7	20	0,841	0,835	0,829
13	2	0,7	15	0,838	0,834	0,829
14	2	0,3	15	0,832	0,830	0,828

continua

conclusão

Mod	Prof. Máx.	Taxa de aprendizagem	nº de árvores	Auroc des	Auroc val	Auroc oot
15	3	0,3	10	0,834	0,830	0,827
16	2	0,5	10	0,832	0,830	0,826
17	2	0,7	10	0,834	0,831	0,826
18	1	0,5	20	0,831	0,832	0,826
19	1	0,3	20	0,826	0,827	0,824
20	1	0,5	15	0,828	0,830	0,823
21	2	0,3	10	0,825	0,824	0,822
22	1	0,7	20	0,830	0,830	0,822
23	1	0,7	15	0,827	0,828	0,820
24	1	0,3	15	0,817	0,819	0,814
25	1	0,5	10	0,820	0,820	0,814
26	1	0,7	10	0,819	0,820	0,813
27	1	0,3	10	0,806	0,807	0,806

Fonte: elaboração própria

Na Tabela 5, constata-se que a utilização de profundidade da árvore igual a 1 diminui significativamente o valor da *auroc*.

4.4. Multilayer perceptron

A análise do método *multilayer perceptron* foi elaborada através dos 24 modelos apresentados na Tabela 6, onde estão ordenados pelo valor das *auroc* da amostra *out of time*. O modelo 1 será comparado aos modelos das outras técnicas por possuir valor de *auroc* igual a 0,832, o maior entre os modelos presentes na Tabela 6.

Tabela 6 - Hiperparâmetros e auroc do *multilayer perceptron*

Mod	Nº max iterações	Neurônios 1ª camada	Neurônios 2ª camada	Auroc des	Auroc val	Auroc oot
1	10	100	100	0,841	0,837	0,832
2	10	100	10	0,839	0,835	0,832
3	5	100	100	0,839	0,836	0,831
4	5	50	50	0,837	0,835	0,831
5	5	100	10	0,837	0,834	0,831
6	10	30	10	0,836	0,833	0,830
7	10	50	10	0,837	0,833	0,830
8	10	50	50	0,836	0,832	0,829

continua

conclusão

Mod	Nº max iterações	Neurônios 1ª camada	Neurônios 2ª camada	Auroc des	Auroc val	Auroc oot
9	1	100	10	0,833	0,832	0,829
10	5	30	30	0,833	0,832	0,829
11	5	50	10	0,836	0,834	0,829
12	1	50	50	0,832	0,830	0,827
13	5	30	10	0,832	0,832	0,827
14	1	50	10	0,830	0,830	0,827
15	1	30	10	0,830	0,830	0,827
16	10	30	30	0,835	0,832	0,826
17	1	30	30	0,829	0,828	0,826
18	5	10	10	0,830	0,829	0,826
19	1	100	100	0,831	0,830	0,825
20	1	10	10	0,826	0,828	0,824
21	10	10	10	0,830	0,827	0,824

Fonte: elaboração própria

Observando o valor da *auroc* da amostra de desenvolvimento, concluímos que não houve sobreajuste.

4.5. Comparação entre os métodos

Será realizado um teste de hipóteses com nível de significância a 5% utilizando o método teste DeLong nas *auroc* dos melhores modelos de cada técnica, i.e., aqueles com maior *auroc* para cada técnica utilizada. A comparação será em relação à *auroc* com o maior valor entre elas, que neste caso é a da técnica *random forests*.

H_0: As áreas sob as curvas roc do método em questão e do modelo 1 da técnica *random forests* são iguais.
H_1: A diferença entre as áreas das duas curvas é diferente de zero.

Os resultados das *auroc*, bem como hiperparâmetros e valor p estão na Tabela 7.

Escolhida por deter o maior valor, a *auroc* da técnica *random forests* foi comparada com as dos outros métodos, como os valores p resultaram em menos do que 0,05, rejeitamos a hipótese nula, i.e., significativamente, as *auroc* não são iguais, a maior *auroc* é a do método *random forests*, com valor igual a 0,836.

Abdou e Pointon (2011) pontua que não existe uma técnica que seja a melhor para todas as bases de dados. Da mesma maneira, Fuhr *et al.* (2017) ressalta que não há superioridade de uma técnica sobre a outra devido às bases de dados, que se alteram dada a situação ou instituição a que pertencem.

Tabela 7 - Comparação com *auroc* do *random forests*

Método	Hiperparâmetros	Auroc oot	Valor p
Regressão logística	nº de vars. Independentes = 61 Variáveis categorizadas	0,833	0,02
Random forests	nº de árvores = 500 nº de variáveis por árvore = 8 nº mínimo de obs por folha = 50	0,836	-
XGBoost	Profundidade máxima = 3 Taxa de aprendizado = 0,3 nº de árvores = 20	0,833	0,02
Multilayer perceptron	nº máximo de iterações = 10 nº neurônios 1ª camada = 100 nº neurônios 2ª camada = 100	0,832	0,00

Fonte: elaboração própria

Portanto, a melhor escolha para essa base de dados será o método no qual o analista esteja mais familiarizado para tirar o melhor proveito e demande menos recursos computacionais.

5. CONCLUSÃO

Em sua grande maioria, credores que emprestam recursos financeiros visam aumentar o lucro e diminuir as perdas, neste cenário, é de fundamental importância a previsão sobre a probabilidade de um possível devedor ser bom ou mau pagador. Além disso, normas regulatórias, como os acordos de Basileia, que tem como objetivo reduzir o risco de uma crise financeira para a economia, devem ser cumpridas para que a instituição de crédito permaneça dentro das exigências estipuladas.

O objetivo era examinar se há um método matemático ou estatístico que seja superior ao prever o escore de crédito de clientes de instituições financeiras brasileiras. Neste trabalho, comparou-se as técnicas regressão logística, comumente utilizada em modelos de escore de crédito, *random forests*, método de ensemble de árvores, *XGBoost*, cujo algoritmo central é o *gradient boosting*, e *multilayer perceptron*, proveniente da classe de algoritmos de rede neural artificial.

A base de dados constituída de pequenas e médias empresas brasileiras de diversos setores da economia foi dividida em três amostras: desenvolvimento, validação e *out of time*. Variáveis que continham valores ausentes foram excluídas e categorizadas por meio do algoritmo de árvores de inferência condicional, as com valor da informação igual a zero também foi excluída.

Comparou-se as áreas sob a curva ROC (*auroc*) de diversos modelos elaborados mediante ajuste de hiperparâmetros das técnica *random forests*, *XGBoost* e *multilayer perceptron*. No caso da regressão logística, os modelos eram produzidos por meio de categorização de variáveis e importância das variáveis. Os com maior *auroc* eram os escolhidos para a comparação efetuada junto aos outros modelos.

Finalmente, os quatro melhores modelos de cada técnica, i.e., com maior *auroc*, foram comparados por meio de um teste de hipóteses utilizando o teste *DeLong*, o que resultou na superioridade do método *random forests*, cuja *auroc* era significantemente maior do que as dos outros métodos apresentados. Porém, cabe ao analista escolher o melhor método ao analisar o custo-benefício, já que alguns métodos requerem menos recursos computacionais e menor tempo de processamento .

Sugestões para pesquisas futuras incluem a utilização de outras técnicas de aprendizado de máquina como *adaboost*, *support vector machine* (SVM), *Bayesian additive regression trees* (BART) e *extreme learning machine* (ELM), a aplicação do

método de validação cruzada (do inglês: *k-fold cross validation*), a técnica *chi-square automatic interaction detector* (CHAID) para categorizar a base de dados e, ainda, ampliar o estresse dos hiperparâmetros utilizados e incluir o uso de outros.

REFERÊNCIAS

ABDOU, H. A.; POINTON, J. Credit scoring, statistical techniques and evaluation criteria: a review of the literature. **Intelligent systems in accounting, finance and management**, v. 18, n. 2-3, p. 59-88., 2011.

ABELLÁN, J.; CASTELLANO, J. G. A comparative study on base classifiers in ensemble methods for credit scoring. **Expert Systems with Applications**, v. 73, p. 1-10, 2017.

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

BANCO CENTRAL DO BRASIL (BCB). **Recomendações de Basileia**. Disponível em: <https://www.bcb.gov.br/fis/supervisao/basileia.asp>. Acesso em: 20 de set. 2018.

BANK FOR INTERNATIONAL SETTLEMENT (BIS). **History of the Basel Committee**. Disponível em: <https://www.bis.org/bcbs/history.htm>. Acesso em: 20 de set. 2018

BEQUÉ, A.; LESSMAN, S. Extreme learning machines for credit scoring: An empirical evaluation. **Expert Systems with Applications**, n. 86, p. 42-53, 2017.

BERGMEIR, C.; BENITEZ, J. M. Neural Networks in R Using the Stuttgart Neural Network Simulator: RSNNS. **Journal of Statistical Software**, v. 46, n. 7, p. 1-26, 2012.

BORGES, R. R. *et al*. Sincronização de disparos em redes neuronais com plasticidade sináptica. **Revista Brasileira de Ensino de Física**, v. 37, n. 2, 2015. Disponível em: http://dx.doi.org/10.1590/S1806-11173721787>. Acesso em: 26 nov. 2018.

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

BREIMAN, L. Bias, variance, and arcing classifiers. **Technical report**, n. 460, 1996a.

BREIMAN, L. *et al*. **Classification and regression trees**. Wadsworth, 1984.

BREIMAN, L. Random forests. **Machine learning**, 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**, v.39, n. 3, p. 3446-3453, 2012.

CHEN, T.; GUESTRIN, C. XGBoost: A scalable tree boosting system. **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, São Francisco, Califórnia, EUA, p. 785-794, 13-17 ago., 2016.

CHEN, T.; HE, T. Xgboost: extreme Gradient Boosting. 2018. Disponível em: <https://CRAN.R-project.org/package=xgboost>. Acesso em: 11 nov. 2018.

CHESSER, D. Prediction loan noncompliance. **Journal of Commercial Bank Lending**, v. 56, n. 8, p. 28-38, 1974.

COX, D. The regression analysis of binary sequences. **Journal of the Royal Statistical Society**, v. 20, n. 2, p. 215-242, 1958.

DELONG, E. R. *et al*. Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. **Biometrics**, v.44, n.3, p. 837-845, 1988.

DIETTERICH, T. G. Ensemble methods in machine learning. **Multiple classifier systems**, v. 1857, p. 1–15, 2000.

DURAND, D. **Risk elements in consumer instalment financing**. NBER Books. 1941.

FAWCETT, T. An introduction to roc analysis. **Pattern recognition letters**, v. 27, n. 8, p. 861–874, 2006.

FISHER, R. A. The use of multiple measurements in taxonomic problems. **Annals of eugenics**, v. 7, n. 2, p. 179-188, 1936.

FREUND, Y; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. **Journal of computer and system sciences**, v. 55, n.1, p. 119-139, 1997.

FREUND, Y.; SCHAPIRE, R. E. A short introduction to boosting. **Journal of Japanese Society for Artificial Intelligence**, v. 14(5), p. 771-780, 1999.

FRIEDMAN, J. Greedy function approximation: a gradient boosting machine. **Annals of statistics**, v. 29, n. 5, p. 1189-1232, 2001.

FRIEDMAN, J. *et al*. **The elements of statistical learning**. Springer series in statistics, v. 1, 2001.

FUHR, F. *et al*. Uma revisão sistemática da literature sobre credit scoring. **VII Congresso brasileiro de engenharia de produção**, 6-8 dez., 2017.

GÉRON, A. **Hands-On machine learning with scikit-learn & TensorFlow**: concepts, tools, and techniques to build intelligent systems. O'Reilly media, 2017.

HANLEY, J. A.; HAJIAN-TILAKI, K. O. Sampling variability of nonparametric estimates of the areas under receiver operating characteristic curves: an update. **Academic radiology**, v. 4, n. 1, p. 49-58, 1997.

HANLEY, J. A.; MCNEIL, B. J. A method of comparing the areas under receiver operating characteristic curves derived from the same cases. **Radiology**, v. 148, n. 3, p. 839-843, 1983.

HEDIBERT, P. Disponível em: <http://hedibert.org/wp-content/uploads/2018/06/CART.pdf>. Acesso em: 13 de nov. 2018.

HOTHORN, T. *et al*. ctree: Conditional inference trees. Cran.r_project. Disponível em: < https://rdrr.io/rforge/partykit/f/inst/doc/ctree.pdf >. Acesso em: 11 nov. 2018.

HOTHORN, T. *et al*. Unbiased recursive partitioning: A conditional inference framework. **Journal of Computational and Graphical Statistics**, 15:3, p. 651-674, 2006.

HUANG, B.; THOMAS, L. C. The impact of Basel Accords on the lender's profitability under different pricing decisions. **Journal of the Operational Research Society**, 66:11, p. 1826-1839, 2015.

HURLEY, M.; ADEBAYO, J. Credit Scoring in the era of Big Data. **Yale Journal of Law and Technology**, v. 18, n.1, p. 147-216, 2016.

JAMES, G. *et al*. **An Introduction to Statistical Learning**. Springer Texts in Statistics, v. 103, 2013.

JIANG, W. On weak base hypotheses and their implications for boosting regression and classification. **The annals of statistics**, v. 30, n. 1, p. 51-73, 2002.

KAGGLE. **A Kaggle master explains gradient boosting**. Disponível em: <http://blog.kaggle.com/2017/01/23/a-kaggle-master-explains-gradient-boosting/>. Acesso em: 20 de set. 2018.

KINGSFORD, C.; SALZBERG, S. L. What are decision trees? **Nature Biotechnology**, v. 26, n. 9, p. 1011-1013, 2008.

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

LEWIS, E. M. **An introduction to credit scoring**. Fair, Isaac and Company. 1992.

LIAW, A.; WIENER, M. Classification and Regression by randomForest. **R News**, v. 2, n. 3, p. 18-22, 2002.

LOUZADA, F. *et al*. On the impact of disproportional samples in credit scoring models: An application to a Brazilian bank data. **Expert Systems with Applications**, v. 39, n. 9, p. 8071-8078, 2012.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, n. 4, p. 115-133, 1943.

MEISEL, W. S.; MICHALOPOULOS, D. A. A Partitioning Algorithm with Application in Pattern Classification and the Optimization of Decision Trees. **IEEE Transactions on Computers**, v. c-22, n. 1, p. 93-103, 1973.

MINER, J. R. Pierre-François Verhulst, the discoverer of the logistic curve. **Human Biology**, v. 5, ed. 4, 1933.

MYERS, J. H.; FORGY, E. W. The development of numerical credit evaluation systems. **Journal of the American Statistical Association**, v. 58, n. 303, p. 799-806, 1963.

MOREIRA, T. Inadimplência de empresas desce a menor nível desde 2014. **Valor econômico**, 27 set. 2018. Disponível em: <https://www.valor.com.br/financas/5887015/inadimplencia-de-empresas-desce-menor-nivel-desde-2014>. Acesso em: 30 de set. 2018.

NATEKIN, A.; KNOLL, A. Gradient boosting machines, a tutorial. **Frontiers in Neurorobotics**, v. 7, 2013.

OPDAL, K. *et al*. Will machine learning and hyperparameter optimization become a game changer for credit scoring? 2017.

ORGLER, Y. E. A credit scoring model for comercial loans. **Journal of Money, Credit and Banking**, v. 2, n. 4, p. 435-445, 1970.

RIBEIRO, M. T. *et al*. "Why should I trust you?": Explaining the predictions of any classifier. **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**, São Francisco, Califórnia, EUA, p. 1135-1144, 13-17 ago., 2016.

ROBIN, X. *et al*. pROC: an open-source package for R and S+ to analyze and compare ROC curves. **BMC Bioinformatics**, v. 12, p. 77, 2011.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, p. 386-408, 1958.

SCHAPIRE, R. E. The boosting approach to machine learning: An overview. **Nonlinear Estimation and Classification**, v. 171, p. 149-171, 2003.

SCHAPIRE, R. E. The strength of weak learnability. **Machine learning**, v. 5, n. 2, p. 197-227, 1990.

SIDDIQI, N. **Credit risk scorecards**: developing and implementing intelligent credit scoring. John Wiley & Sons, 2012.

SIGACT. **Prizes**. Gödel Prize. Disponível em: <https://www.sigact.org/prizes/g%C3%B6del/2003.html>. Acesso em: 20 nov. 2018.

STRASSER, H.; WEBER, C. On the asymptotic theory of permutation statistics. **Adaptive Information Systems and Modelling in Economics and Management Science**, v. 8, n. 27, p. 220-250, 1999.

THOMAS, L. C. *et al*. A survey of the issues in consumer credit modelling research. **Journal of the Operational Research Society**, v. 56, n.9, p. 1006-1015, 2005.

THOMAS, L. C. **Consumer Credit Models: Pricing, Profit and Portfolios**. OUP Oxford, 2009.

THOMAS, L. C. *et al*. **Credit scoring and its applications**. Siam, 2002.

THOMAS, L. C. *et al*. **Credit scoring and its applications**. 2º ed. Filadélfia: Society for Industrial and Applied Mathematics, p. 265-305, 2017.

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

APÊNDICES

APÊNDICE A - DICIONÁRIO DE APRENDIZADO DE MÁQUINA

Amostra de desenvolvimento ou treino (*training set*): amostra utilizada para treinar o modelo, i.e., os dados de entrada dessa amostra são pareados com os dados de saída de modo a desenvolver os coeficientes do modelo.

Amostra de Validação (*validation set*): amostra utilizada para mensurar a predição do modelo enquanto os hiperparâmetros são ajustados.

Amostra de teste (*test set*): amostra que fornece a performance final do modelo, diferente da amostra de validação, essa não é viesada nos hiperparâmetros.

Amostra fora da amostra (*out of sample*): o mesmo que amostra de validação ou teste.

Amostra fora do tempo (*out of time*): o mesmo que amostra de validação ou teste, porém compreende um período diferente do período da amostra de desenvolvimento.

Aprendizado supervisionado (*supervised learning*): é um tipo de algoritmo que necessita da variável resposta na amostra de dados para predizer a resposta.

Aprendizado não supervisionado (*unsupervised learning*): é um tipo de algoritmo que não necessita da variável resposta na amostra de dados para predizer a resposta. Em geral, esses algoritmos classificam as observações da amostra em grupos.

Base de dados balanceada (*balanced dataset*): base de dados cujas classes possuem a mesma proporção de observações, por exemplo 50% bons pagadores e 50% maus. Alguns métodos necessitam desse tipo de base para atingir melhores predições, caso contrário precisam passar por alguns ajustes ou tratamento da base.

Base de dados desbalanceada (*unbalanced dataset*): base de dados cujas classes não possuem a mesma proporção de observações, por exemplo 90% bons pagadores e 10% maus.

Big data: termo que descreve o grande volume de dados que podem ser analisados computacionalmente com o intuito de encontrar padrões, tendências e associações entre características de um determinado objetivo.

Hiperparâmetros (ou meta-parâmetros): são parâmetros definidos antes do processo de treino. Cada algoritmo possui seu próprio conjunto de hiperparâmetros. Esses hiperparâmetros são tunados de modo a aumentar o potencial de previsão da resposta do algoritmo.

Métodos de aprendizado de máquina: são algoritmos elaborados com base em técnicas matemáticas e estatísticas com intuito de predizerem uma resposta para uma

observação a partir de suas características. Esses algoritmos “aprendem” a partir da amostra de dados.

Métodos de *ensemble* (*ensemble methods*): são métodos que combinam um conjunto de preditores fracos para aumentar o potencial preditivo.

Preditores fracos (*weak learners*): são preditores cujo poder de predição é um pouco maior do que a previsão aleatória.

Sobreajuste (*overfitting*): é quando o modelo se ajusta perfeitamente a amostra de dados de desenvolvimento se tornando ineficaz para prever os resultados em amostras que contenham outras observações.

Taxa de aprendizado (*learning rate or shrinkage*): hiperparâmetro presente em diversos métodos, consiste em agregar um multiplicador que controla o ajuste dado aos coeficientes (pesos) do algoritmo para evitar o sobreajuste do modelo.

Treinar preditor: processo entre o algoritmo e a amostra de dados visando o desenvolvimento de um modelo com boa predição.

Tunar/tunning: é o ato de ajustar os hiperparâmetros do algoritmo de modo a aumentar a performance do modelo.

Votação (*voting*): é a escolha da classificação com maior ocorrência dentro do espaço estudado.

APÊNDICE B - DESCRIÇÃO DAS VARIÁVEIS DA AMOSTRA

Tabela 8 - Descrição das variáveis

Variável	Descrição	Utilizada
TARGET	Variável Resposta	✓
BASE	Indicador de amostra	✓
DT_T0	Safra de Referência	✓
DiasFund	Idade (em dias) da empresa	✓
TpEntrUltSoc	TpEntrUltSoc	✓
VP17PerUltSemRestr	Periodo de tempo em que ficou pela última vez sem dívida vencida negativada, desde a primeira inclusão de dívida vencida (negativada ou regularizada)	
VP17QtCCFsIncl90d	Quantidade de cheques sem fundos vencidos (negativados ou regularizados) incluídos nos últimos 90 dias.	
VP17QtMaxRestrAti	Quantidade máxima de dívidas vencidas que ficaram negativadas ao mesmo tempo, desde a primeira inclusão de dívida vencida (negativada ou regularizada)	✓
VP17QtPefinsAti030DPI	Quantidade de dívidas Serasa (PEFIN) vencidas negativadas incluídas nos últimos 30 dias	
VP17QtProtInc60d	Quantidade de protestos vencidos (negativados ou regularizados) incluídos nos últimos 60 dias.	

Variável	Descrição	Utilizada
VP17QtRefinsATI	Quantidade de dívidas Serasa (REFIN) vencidas negativadas	
VP17QtRefinsAti030DPI	Quantidade de dívidas Serasa (REFIN) vencidas negativadas incluídas nos últimos 30 dias	
VP17QtRefinsResU6m	Quantidade de dívidas Serasa (REFIN) vencidas regularizadas nos últimos 180 dias	✓
VP17QtRestAtiGBcoOut360d	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas negativadas, incluídas pelo grupo dos Bancos de Pequeno e Médio Porte nos últimos 360 dias	
VP17QtRestAtiGBcoPri360d	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas negativadas, incluídas pelo grupo dos maiores Bancos Privados nos últimos 360 dias	
VP17QtRestrOrigBCOAti030DPI	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas negativadas, incluídas nos últimos 30 dias por Bancos Múltiplos	
VP17QtRestrOrigFINAti	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas negativadas, incluídas por Financeiras	
VP17QtRestrOrigOUTAti030DPI	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas negativadas, incluídas por empresas de outros segmentos diferentes de Bancos, Varejo, Financeiras, Telefonias, Utilities e Seguradoras nos últimos 30 dias	
VP17QtRestrOrigSEG	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas (negativadas ou regularizadas) incluídas por Seguradoras	
VP17QtRestrOrigSEGResA6m	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas regularizadas a mais de 180 dias, incluídas por Seguradoras	
VP17QtRestrOrigSFNInc2Ares	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas regularizadas, incluídas por Bancos (Múltiplos, Comerciais, de Investimentos, Cooperativos, de Desenvolvimento e Caixas Econômicas) nos últimos 2 anos.	✓

Variável	Descrição	Utilizada
VP17QtRestrOrigTELECOMInc60d	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas (negativadas ou regularizadas) incluídas por empresas de Telefonia nos últimos 60 dias.	
VP17QtRestrOrigTELECOMInc90d	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas (negativadas ou regularizadas) incluídas por empresas de Telefonia nos últimos 90 dias.	
VP17QtRestrOrigVARAti	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas negativadas, incluídas por empresas do setor de Comércio	
VP17Qt_ConsSSG_360d	Quantidade de consultas (exceto de Seguradoras) nos últimos 360 dias	
VP17Qt_ConsTOD_720d	Quantidade de consultas (de todos os segmentos) nos últimos 720 dias	✓
VP17TpMed_ConsSTS_30d	Tempo médio entre consultas (exceto de Telecoms e Seguradoras) nos últimos 30 dias	
VP17TpPri_ConsSFN_5ª	Tempo desde a primeira consulta do setor financeiro nos últimos 5 anos	✓
VP17TpPri_ConsSSG_5ª	Tempo desde a primeira consulta (exceto de Seguradoras) nos últimos 5 anos	✓
VP17TpPri_ConsSTS_5ª	Tempo desde a primeira consulta (exceto de Telecoms e Seguradoras) nos últimos 5 anos	✓
VP17TpPri_ConsTOD_5ª	Tempo desde a primeira consulta (de todos os segmentos) nos últimos 5 anos	✓
VP18QtRestrAti	Quantidade de dívidas vencidas negativadas	

Variável	Descrição	Utilizada
VP18QtRestrSemTELECOMRes180d	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas regularizadas nos últimos 180 dias, incluídas por empresas diferentes de Telefonia.	
VP18Qt_ConsTOD_360d	Quantidade de consultas (de todos os segmentos) nos últimos 360 dias	✓
VP18TpRestrMaxAti	Tempo desde a regularização da dívida vencida regularizada com maior período em que ficou negativada	✓
VPerMaxSemRestr	Periodo de tempo máximo em que ficou sem dívida vencida negativada, desde a primeira inclusão de dívida vencida (negativada ou regularizada)	✓
VPerMedSemRestr2	Periodo de tempo médio em que ficou sem dívida vencida negativada, desde a primeira inclusão de dívida vencida (negativada ou regularizada)	✓
VQtAcoesAti	Quantidade de ações (todos os tipos) vencidas negativadas	
VQtCredoresTot	Número de diferentes empresas que incluíram dívidas Serasa (REFIN e PEFIN) vencidas (negativadas ou regularizadas)	✓
VQtMaxRestrAti	Quantidade máxima de dívidas vencidas que ficaram negativadas ao mesmo tempo, desde a primeira inclusão de dívida vencida (negativada ou regularizada)	✓
VQtPefinsATI	Quantidade de dívidas Serasa (PEFIN) vencidas negativadas	✓
VQtProtAti	Quantidade de protestos vencidos negativados	✓
VQtProtInc3A	Quantidade de protestos vencidos (negativados ou regularizados) incluídos nos últimos 3 anos.	✓

Variável	Descrição	Utilizada
VQtProtInc4A	Quantidade de protestos vencidos (negativados ou regularizados) incluídos nos últimos 4 anos.	✓
VQtRefinsATI	Quantidade de dívidas Serasa (REFIN) vencidas negativadas	
VQtRestrExc60d	Quantidade de dívidas vencidas regularizadas nos últimos 60 dias.	✓
VQtRestrInc090d	Quantidade de dívidas vencidas (negativadas ou regularizadas) incluídas entre 31 e 90 dias	✓
VQtRestrInc1A	Quantidade de dívidas vencidas (negativadas ou regularizadas) incluídas no último ano.	✓
VQtRestrInc3A	Quantidade de dívidas vencidas (negativadas ou regularizadas) incluídas nos últimos 3 anos.	✓
VQtRestrIncU180d	Quantidade de dívidas vencidas (negativadas ou regularizadas) incluídas nos últimos 180 dias.	✓
VQtRestrIncU60d	Quantidade de dívidas vencidas (negativadas ou regularizadas) incluídas nos últimos 60 dias.	✓
VQtRestrOrigSFNInc2A	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas (negativadas ou regularizadas) incluídas por Bancos (Múltiplos, Comerciais, de Investimentos, Cooperativos, de Desenvolvimento e Caixas Econômicas) nos últimos 2 anos.	✓
VQtRestrOrigSFNInc2Ares	Quantidade de dívidas Serasa (REFIN e PEFIN) vencidas regularizadas, incluídas por Bancos (Múltiplos, Comerciais, de Investimentos, Cooperativos, de Desenvolvimento e Caixas Econômicas) nos últimos 2 anos.	✓
VQtRestrResAtraso30dU12m	Quantidade de dívidas vencidas regularizadas no último ano com atraso superior a 30 dias (regularizadas a mais de 30 dias do vencimento)	✓

Variável	Descrição	Utilizada
VQtRestrTot	Quantidade de dívidas vencidas (negativadas ou regularizadas)	✓
VQt_ConsFIN_5d	Quantidade de consultas de Financeiras nos últimos 5 dias	✓
VQt_ConsFIN_720d	Quantidade de consultas de Financeiras nos últimos 720 dias	✓
VQt_ConsSFN_720d	Quantidade de consultas do setor financeiro nos últimos 720 dias	✓
VQt_ConsTOD_360d	Quantidade de consultas (de todos os segmentos) nos últimos 360 dias	✓
VQt_ConsTOD_5ª	Quantidade de consultas (de todos os segmentos) nos últimos 5 anos	✓
VQt_ConsTOD_720d	Quantidade de consultas (de todos os segmentos) nos últimos 720 dias	✓
VQt_ConsTOD_90d	Quantidade de consultas (de todos os segmentos) nos últimos 90 dias	✓
VTdRel_ConsSTS_P05dU75d	Tendência relativa de consultas (exceto de Telecoms e Seguradoras) em períodos de 5 dias nos últimos 75 dias	✓
VTpMed_ConsSTS_180d	Tempo médio entre consultas (exceto de Telecoms e Seguradoras) nos últimos 180 dias	✓
VTpMed_ConsSTS_360d	Tempo médio entre consultas (exceto de Telecoms e Seguradoras) nos últimos 360 dias	✓

Variável	Descrição	Utilizada
VTpMed_ConsSTS_720d	Tempo médio entre consultas (exceto de Telecoms e Seguradoras) nos últimos 720 dias	✓
VTpPriRestrIncAti	Tempo desde a primeira inclusão de dívida vencida negativada	✓
VTpPri_ConsBCO_5ª	Tempo desde a primeira consulta de Bancos Múltiplos nos últimos 5 anos	✓
VTpPri_ConsCFL_5ª	Tempo desde a primeira consulta para comércio e financiamento de veículos leves (carros e motos) nos últimos 5 anos	✓
VTpPri_ConsCHQSTS_5ª	Tempo desde a primeira consulta à cheques (exceto Telecoms e Seguradoras) nos últimos 5 anos	✓
VTpPri_ConsCRDSTS_5ª	Tempo desde a primeira consulta aos relatórios de crédito (exceto Telecoms e Seguradoras) nos últimos 5 anos	✓
VTpPri_ConsCTV_5ª	Tempo desde a primeira consulta de Adm. de Cartões de Crédito exclusivamente fornecidos pelo Varejo nos últimos 5 anos	
VTpPri_ConsIND_5ª	Tempo desde a primeira consulta de Indústrias nos últimos 5 anos	✓
VTpPri_ConsSFN_5ª	Tempo desde a primeira consulta do setor financeiro nos últimos 5 anos	✓
VTpPri_ConsTCO_5ª	Tempo desde a primeira consulta de Telecoms (inclusive fornecedoras de sinal de tv a cabo e internet) nos últimos 5 anos	✓
VTpPri_ConsTOD_5ª	Tempo desde a primeira consulta (de todos os segmentos) nos últimos 5 anos	✓

Variável	Descrição	Utilizada
VtpQtMaxRestrAti	Tempo desde o período com a quantidade máxima de dívidas vencidas que ficaram negativadas ao mesmo tempo, desde a primeira inclusão de dívida vencida (negativada ou regularizada)	✓
VtpRestrMaxAti	Tempo desde a regularização da dívida vencida regularizada com maior período em que ficou negativada	✓
VtpUltProtInc	Tempo desde a última inclusão de protesto vencido (negativado ou regularizado)	✓
VtpUltRestrInc	Tempo desde a ultima inclusão de dívida vencida (negativada ou regularizada)	✓
VtpUltRestrRes	Tempo desde a ultima regularização de dívida vencida regularizada	✓
VTpUltRestrSemTELECOMInc	Tempo desde a última dívida Serasa (REFIN e PEFIN) vencida (negativada ou regularizada) incluída por empresas diferentes de Telefonia	✓
VtpUltSerasaInc	Tempo desde a última inclusão de dívida Serasa (REFIN e PEFIN) vencida (negativada ou regularizada)	✓
VtpUltSerasaRes	Tempo desde a última regularização de dívida Serasa (REFIN e PEFIN) vencida regularizada	✓
VtxPgtoRestr	Taxa (percentual) do número de dívidas vencidas regularizadas pelo número de dívidas vencidas (negativadas ou regularizadas)	✓
VTxPgtoRestrInc180d	Taxa (percentual) do número de dívidas vencidas regularizadas pelo número de dívidas vencidas (negativadas ou regularizadas) incluídas nos últimos 180 dias	✓

Fonte: elaboração própria

APÊNDICE C - VALOR DA INFORMAÇÃO

Avaliar o poder de predição de cada variável individualmente e agrupá-las em classes são as duas principais tarefas para análise de variável, segundo Siddiqi (2012). Entre as vantagens de agrupá-las estão:

- Facilidade em lidar com *outliers* e classes raras
- Facilita no entendimento da relação entre as variáveis e a performance
- Pode modelar linearmente casos de dependências não lineares
- Permite controle no desenvolvimento dos ajustes dos grupos
- Permite ao usuário desenvolver *insights* sobre os preditores, o que colabora no desenvolvimento de estratégias

Valor da informação (IV, do inglês: *information value*) é uma técnica muito utilizada para selecionar variáveis importantes para um modelo e agrupá-las. Para calcular o IV, utiliza-se o peso da evidência (WOE, do inglês: *weight of evidence*), que consiste em predizer o poder da variável independente em relação à variável resposta. O peso da evidência para cada categoria da variável é calculado da seguinte forma:

$$woe_i = \ln\left(\frac{distr\ bons_i}{distr\ maus_i}\right)$$

onde, *i* é a i-ésima categoria da variável.

O cálculo do valor da informação de cada variável é dado por:

$$IV = \sum_{i=1}^{n} (dist\ bons_i - dist\ maus_i) * woe_i$$

Variáveis com IV entre 0,1 a 0,3 são consideradas com poder preditivo médio, já entre 0,3 e 0,5 são consideradas com poder preditivo forte, abaixo desses valores são consideradas fracas, acima são consideradas suspeitas, como pode ser visto na Tabela 9:

Tabela 9 - Poder de predição do valor da informação

IV	Poder preditivo
< 0,02	inútil
0,02 a 0,1	fraco
0,1 a 0,3	médio
0,3 a 0,5	forte
> 0,5	suspeito

Fonte: elaboração própria

Os modelos deste trabalho foram produzidos com as 61 variáveis da Tabela 10.

Tabela 10 - Valor da informação das variáveis

	Variáveis	IV	Média ou forte
1	VQtRestrInc1A	1,10	
2	VQtRestrIncU180d	1,04	
3	VTpUltRestrInc	1,04	
5	VQtRestrInc3A	0,94	
4	VTxPgtoRestrInc180d	0,93	
6	VQtMaxRestrAti	0,88	
7	VQtRestrIncU60d	0,82	
8	VQtRestrTot	0,78	
9	VQtRestrInc090d	0,67	
10	VPerMedSemRestr2	0,65	
12	VTpUltRestrRes	0,62	
14	VQtRestrResAtraso30dU12m	0,61	
11	VTpPriRestrIncAti	0,60	
13	VTxPgtoRestr	0,59	
15	VTpUltProtInc	0,53	
16	VQtProtInc3A	0,53	
17	VQtProtInc4A	0,52	
23	VTpUltSerasaInc	0,52	
18	VTpUltRestrSemTELECOMInc	0,48	✓
20	VQtCredoresTot	0,44	✓
21	VPerMaxSemRestr	0,44	✓
19	VTpUltSerasaRes	0,43	✓
22	VQtRestrExc60d	0,41	✓
27	VTpMed_ConsSTS_360d	0,30	✓
24	VQt_ConsSFN_720d	0,30	✓
26	VTpMed_ConsSTS_720d	0,30	✓
28	VQt_ConsTOD_360d	0,29	✓
29	VQt_ConsTOD_720d	0,28	✓

	Variáveis	IV	Média ou forte
30	VTpMed_ConsSTS_180d	0,28	✓
25	VqtProtAti	0,26	✓
32	VQt_ConsTOD_90d	0,25	✓
31	VQtRestrOrigSFNInc2A	0,22	✓
33	VQtRestrOrigSFNInc2ARes	0,21	✓
37	VQt_ConsTOD_5a	0,20	✓
34	VTpQtMaxRestrAti	0,20	✓
40	TpEntrUltSoc	0,19	✓
35	VP17QtRestrOrigSFNInc2ARes	0,18	✓
38	VP17QtRefinsResU6m	0,18	✓
39	VP17QtMaxRestrAti	0,18	✓
36	VQtPefinsATI	0,17	✓
41	VTpRestrMaxAti	0,16	✓
42	VTdRel_ConsSTS_P05dU75d	0,16	✓
46	DiasFund	0,15	✓
43	VQt_ConsFIN_720d	0,14	✓
44	VQt_ConsFIN_5d	0,13	✓
47	VTpPri_ConsBCO_5a	0,10	✓
45	VTpPri_ConsSFN_5a	0,10	
49	VP17TpPri_ConsSTS_5a	0,09	
48	VP17Qt_ConsTOD_720d	0,09	
51	VP17TpPri_ConsSSG_5a	0,08	
52	VP17TpPri_ConsSFN_5a	0,08	
50	VP17TpPri_ConsTOD_5a	0,08	
54	VTpPri_ConsCRDSTS_5a	0,08	
53	VTpPri_ConsCFL_5a	0,07	
55	VTpPri_ConsCHQSTS_5a	0,06	
56	VTpPri_ConsTCO_5a	0,06	
58	VP18Qt_ConsTOD_360d	0,06	
57	VTpPri_ConsIND_5a	0,05	
59	VP18TpRestrMaxAti	0,04	
60	VTpPri_ConsTOD_5a	0,04	
61	DT_T0	0,00	

Fonte: elaboração própria

APÊNDICE D - CATEGORIZAÇÃO POR ÁRVORE DE INFERÊNCIA CONDICIONAL

A categorização de variáveis para implementação nos modelos foi feita com o pacote *smbinning* do R (HOTHORN *et al.*, 2006) por meio do algoritmo árvore de inferência condicional (do inglês: *conditional inference tree*), que se baseia em criar árvores de decisão por partição recursiva (do inglês: *recursive partitioning*) e discretização supervisionada (do inglês: *supervised discretization*), este algoritmo classifica os dados da variável independente em questão de maneira ótima, considerando a variável dependente com a finalidade de diminuir a complexidade dos dados e facilitar o cálculo do algoritmo do método que será implementado. Hothorn *et al.* (2006) apresentou uma estrutura unificada para o particionamento recursivo binário, mensurando a associação entre resposta e covariáveis, o critério de parada é realizado por meio de testes de hipóteses. Para realizar este trabalho, há somente uma variável por árvore, portanto a eficiência do algoritmo será utilizada para particionar cada nó, realizada pela estrutura de teste de permutação (do inglês: *permutation test framework*) que consiste em testar uma hipótese nula aleatória (STRASSER; WEBER, 1999), e para o critério de parada, que acontece quando a hipótese nula de independência entre a resposta e a variável independente não pode ser rejeitada a um certo nível de significância. Outra característica compreendida pelo algoritmo é a exclusão de *missing* no início do processo, que retornam ao final, sendo adicionados na melhor partição conforme o valor da informação.

APÊNDICE E - ÁREA SOB A CURVA ROC

Mensurar o desempenho do modelo é imprescindível para provar sua superioridade em relação à estimativa aleatória e, também, é uma maneira de constituir um comparativo entre modelos, Hanley e McNeil (1983) expuseram a importância do uso de um critério estatístico formal para conjecturar se as diferenças nas acurácias entre métodos são reais ou aleatórias. Neste trabalho, a técnica utilizada para avaliar a performance dos modelos será a área sob a curva roc (auroc, do inglês: *area under the curve roc*).

Para melhor entendimento dessa métrica, Fawcett (2006) explora a matriz de confusão com duas classes (Tabela 11) expondo alguns conceitos:

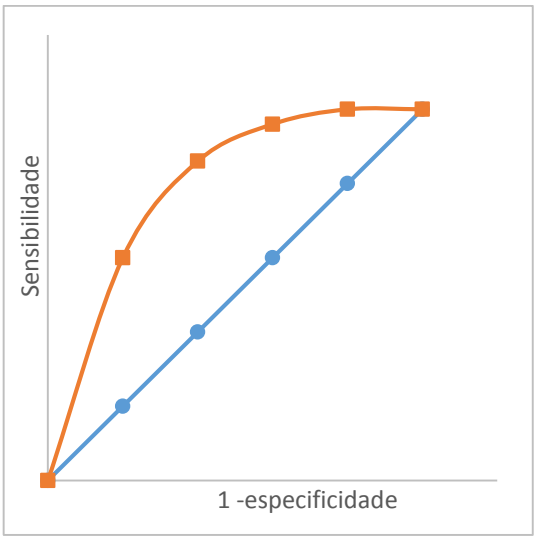
Tabela 11 - Matriz de confusão

		predito	
		positivo	**negativo**
observado	**positivo**	verdadeiro positivo (VP) sensibilidade	falso negativo (FN)
	negativo	falso positivo (FP) 1 - especificidade	verdadeiro negativo (VN) especificidade

Fonte: elaboração própria

onde, sensibilidade = VP/(VP + FN) e (1 - especificidade) = FP/(FP + VN)

Figura 7 - Curva roc

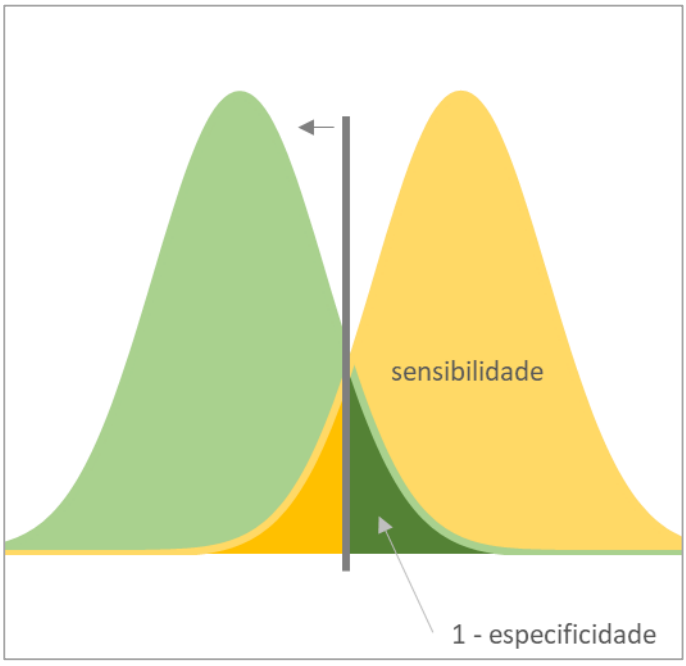


Fonte: elaboração própria

A Figura 7 mostra a curva da estimativa aleatória em azul (marcador redondo) e a curva roc em laranja (marcador quadrado), a área abaixo da linha azul é de 0,5, já abaixo a linha laranja é a auroc, que quanto maior, maior será o poder de predição do modelo.

Constrói-se duas normais, ou outra distribuição, uma para cada classe

Figura 8 - Normais de cada classe



Fonte: elaboração própria

Cada ponto da curva roc na Figura 7 será construído a partir dos valores das áreas de (1 - especificidade) e sensibilidade, contidos no intervalo [0,1], conforme o ponto de corte (linha cinza) se desloca da extrema direita para a esquerda na Figura 8.

Há várias técnicas para fazer a comparação entre *auroc*, a empregada neste trabalho será a *DeLong* (DELONG *et al.*, 1988), que é utilizada para estimar a covariância entre *auroc*, mas pode também calcular a variância de apenas uma *auroc*, de maneira limpa e elegante como mencionado por Hanley e Hajian-Tilaki (1997).

O método não paramétrico *Delong* (DELONG *et al.*, 1988) compara as áreas abaixo das curvas roc explorando propriedades do teste U de Mann-Whitney, que se fundamenta nos postos dos valores ordenados das curvas roc.

Neste trabalho, foi utilizado o pacote do R pRoc (ROBIN *et al.*, 2011).

APÊNDICE F - PACOTES DO R

As técnicas utilizadas neste trabalho foram implementadas com macros já existentes nos pacotes do software R. Os principais pacotes utilizados foram:

1. **smbinning** (*smbinning*): categorização das variáveis da base de dados e valor da informação.
   Disponível em: <https://cran.r-project.org/web/packages/smbinning/smbinning.pdf>

2. **smbinning** (*glm*): regressão logística

3. **randomForest** (*randomForest*): random forests.
   Disponível em: <https://cran.r-project.org/web/packages/randomForest/randomForest.pdf>

4. **xgboost** (*xgboost*): xgboost.
   Disponível em: <https://cran.r-project.org/web/packages/xgboost/xgboost.pdf>

5. **RSNNS** (*mlp*): multilayer perceptron.
   Disponível em: <https://cran.r-project.org/web/packages/RSNNS/RSNNS.pdf>

6. **pROC** (*roc*): área sob a curva roc.
   Disponível em: <https://cran.r-project.org/web/packages/pROC/pROC.pdf>

7. **pROC** (*roc.test*): comparação entre auroc