Técnicas de classificação aplicadas a *credit scoring: revisão sistemática e comparação*

Renato Frazzato Viana

Renato Frazzato Viana

Técnicas de classificação aplicadas a *credit scoring: revisão sistemática e comparação*

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Estatística. *Versão revisada*

Área de Concentração: Estatística

Orientador: Prof. Dr. Francisco Louzada Neto

USP – São Carlos
Fevereiro de 2016

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

F616t

Frazzato Viana, Renato
Técnicas de classificação aplicadas a credit scoring: revisão sistemática e comparação / Renato Frazzato Viana; orientador Francisco Louzada Neto. -- São Carlos, 2015.
113 p.

Dissertação (Mestrado - Programa Interinstitucional de Pós-graduação em Estatística) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2015.

1. Credit scoring. 2. Data mining. 3. Risco de crédito. 4. revisão. 5. técnicas. I. Louzada Neto, Francisco, orient. II. Título.

Renato Frazzato Viana

Classification techniques applied to credit scoring: a systematic review and comparison

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação - ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Statistics. *Final Version.*

Concentration Area: Statistics

Advisor: Prof. Dr. Francisco Louzada Neto

USP – São Carlos
February 2016

Dedico este trabalho aos meus pais Marta e Roberto

Agradecimentos

Agradeço a Deus por mais esta conquista.

Agradeço ao Professor Neto pelos conselhos, ajuda e paciência durante este período.

Agradeço à CAPES pelo auxílio financeiro durante o mestrado.

Agradeço todos os docentes do departamento de Estatística da UFSCar e do ICMC-USP pelo conhecimento fornecido que, sem dúvida nenhuma, levarei para resto da vida.

Agradeço aos meus pais e minha irmã pelo incentivo e motivação durante esse percurso.

Agradeço à Professora Socorro Rangel da UNESP de São José do Rio Preto pelos ensinamentos durante os primeiros passos na iniciação científica.

Agradeço aos companheiros de mestrado pelos bons momentos de estudo e, em especial, ao Victor Hugo e Victor Sung pelas boas conversas.

Resumo

Com a crescente demanda por crédito é muito importante avaliar o risco de cada operação desse tipo. Portanto, ao fornecer crédito a um cliente é necessário avaliar as chances do cliente não pagar o empréstimo e, para esta tarefa, as técnicas de *credit scoring* são aplicadas. O presente trabalho apresenta uma revisão da literatura de *credit scoring* com o objetivo de fornecer uma visão geral das várias técnicas empregadas. Além disso, um estudo de simulação computacional é realizado com o intuito de comparar o comportamento de várias técnicas apresentadas no estudo.

Palavras-Chave: Credit-Scoring, Data-Mining, Risco de Crédito, Revisão, Técnicas.

Abstract

Nowadays the increasing amount of bank transactions and the increasing of data storage created a demand for risk evaluation associated with personal loans. It is very important for a company has a very good tools in credit risk evaluation because theses tools can avoid money losses.

In this context, it is interesting estimate the default probability for a customers and, the credit scoring techniques are very useful for this task. This work presents a credit scoring literature review with and aim to give a overview covering many techniques employed in credit scoring and, a computational study is accomplished in order to compare some of the techniques seen in this text.

Keywords: Credit-Scoring, Data-Mining, Review, Techniques.

Sumário

Capítulo 1

INTRODUÇÃO E REVISÃO SISTEMÁTICA

A necessidade da análise de crédito nasceu no início do comércio em conjunto com as atividades de empréstimo de dinheiro. Entretanto, os conceitos modernos e ideias de *credit scoring* emergiram aproximadamente há 70 anos atrás com Durant(1940). Desde então, empresas têm armazenado informações de crédito para que a decisão de conceder ou não o crédito seja tomada de modo mais seguro (Banasik et al., 1999; Marron, 2007; Louzada et al., 2012).

Segundo Thomas et al. (2002) *credit scoring* é um conjunto de modelos de decisão e suas técnicas que auxiliam credores no momento da concessão de crédito. Nesse trabalho é considerada uma definição mais ampla: *credit scoring* é uma expressão numérica que informa o risco de conceder crédito a um determinado cliente, uma ferramenta útil na prevenção de *default*, métodos importantes na avaliação do risco de crédito, e a atividade de pesquisa na área de gerenciamento de risco de crédito.

Técnicas estatísticas e de mineração de dados têm dado contribuições significativas no campo da ciência da informação e são capazes de construir modelos para medir o risco de um cliente condicionado em suas características e, classificar este cliente como bom ou mau pagador de acordo com o seu nível de risco. Por isso, a ideia principal dos modelos de *credit scoring* é identificar características que influenciam no comportamento do cliente, sendo que a classificação ocorre dentro de dois grupos distintos caracterizados pela decisão de aceitar ou rejeitar a aplicação de crédito (Han et al. 2006).

A tentativa de gerenciar o risco tem feito com que as instituições buscassem uma melhoria nas técnicas usadas para risco de crédito. Esse fato resultou no desenvolvimento e aplicação de vários modelos quantitativos nesse cenário. Entretanto, em muitos casos a escolha do modelo está relacionada com subjetividade do analista ou, ainda, sendo o método escolhido o mais

recente possível. Existem também propriedades que normalmente diferem, tais como o número de base de dados aplicadas para verificar a qualidade da performance do modelo ou até mesmo outro tipo de validação e procedimentos para avaliar o custo de classificações errôneas.

Existem também outros eventos naturais, uma vez que *credit scoring* tem sido usado em várias outras áreas, incluindo a proposta de novos métodos ou a comparação entre técnicas diferentes utilizadas em problemas de predição e classificação. Esse trabalho faz uma revisão bibliográfica sistemática das técnicas de classificação utilizadas em *credit scoring* e, além disso, apresenta um estudo de simulação.

Embora a série de artigos é focada na mesma área, muitos deles têm objetivos distintos e, é possível separá-los em objetivos gerais similares. Esse trabalho tem como objetivos fazer um levantamento sobre: as medidas de performance mais comuns; as técnicas mais comuns que foram utilizadas no problema de *credit scoring* e como a utilização das técnicas vem mudando com o passar do tempo através do surgimento de novas técnicas. Além disso, algumas outras informações foram coletadas como, por exemplo, ano de publicação, autor e revista. Um estudo desse tipo fornece contribuições no entedimento amplo da literatura de *credit scoring* com relação as técnicas de utilizadas.

Alguns autores propuseram alguns métodos de classificação. Lee et al. (2002) introduziram *neural discriminant model*, Gestel et al. (2006) propuseram a *support vector machine model* com uma a abordagem Bayesiana. Hoffman et al. (2007) propuseram o *boosted genetic fuzzy model*. Hsieh and Hung. (2010) utilizaram um método combinado que abordou rede neural, *support vector machine* e redes bayesianas. Shi. (2010) fez uma revisão sistemática da literatura que abrangeu múltiplos critérios da programação linear aplicado ao *credit scoring* de 1969 até 2010. Outras revisões da literatura foram feitas por Hand e Henley (1997); Gemela (2001); Xu et al (2009); Shi (2010); Lahsasma et al (2010); Van Gool et al (2012).

Entre os artigos que fizeram uma discussão conceitual, Bardos. (1998) apresentou ferramentas usadas pelo Banque de France. Banasik et al. (1999) discutiram como *hazard models* podem ser considerados para investigar quando os clientes irão entrar em *default*, Hand. (2001) discutiu as aplicações e desafios na análise de *credit scoring*. Martens et al. (2010) realizaram uma aplicação do *credit scoring* e discutiram como suas ferramentas se ajustam no sistema global de gerenciamento de crédito Basiléia II. Outros exemplos sobre discussão conceitual podem ser vistos em Chen and Huang .(2003); Marron. (2007); Thomas. (2010).

Comparando técnicas tradicionais, West. (2000) comparou cinco modelos de redes neurais com técnicas mais conhecidas. Os resultados indicaram que rede neural possui uma boa acurácia em *credit scoring* e também, que a regressão logística é uma boa alternativa à redes neurais. Baesens et al. (2003) fizeram uma comparação envolvendo análise de discriminante, regressão

logística, *support vector machine*, neural network, redes bayesianas, árvores de decisão e *k-nearest neighbors* e concluiram que várias técnicas de classificação possuem uma performance que são bastante próximas. Outras comparações importantes podem ser vistas em Adams et al. (2001); Hoffmann et al. (2002); Ong et al. (2005); Baesens et al. (2005); Wang et al. (2005); Lee et al. (2006); Huang et al. (2006); Xiao et al. (2006); Van Gestel et al.(2007); Martens et al. (2007); Hu and Ansell. (2007); Tsai. (2008); Abdou et al. (2008); Sinha and Zhao. (2008); Luo et al. (2009); Finlay. (2009); Abdou. (2009); Hu and Ansell. (2009); Finlay. (2010); Wang et al. (2011); Também, Liu and Schumann. (2005); Somol et al. (2005); Tsai (2009); Falangis and Glen. (2010); Chen and Li. (2010); Yu and Li. (2011); McDonald et al (2012); Wang et al. (2012); trabalharam com seleção de covariáveis. Hand and Henley. (2007); Gemela (2001); Xu et al. (2009); Shi. (2010); Lahsasna et al. (2010); Van Gool et al. (2012) produziram uma revisão literária. Yang et al. (2004); Hand. (2005); Lan et al. (2006); Dryver and Sukkasem. (2009) trabalharam com medidas de performance. Existem outros artigos que abordaram outros assuntos tais: seleção de modelos (Ziari et al. 1997), impacto da amostra (Verstraem and van Den Poel, 2005), segmentação e acurácia (Bijak and Thomas, 2012).

A revisão bibliografica sistemática, também conhecida como revisão sistemática, é uma alternativa adequada para identificar e classificar contribuições científicas na área e descrever de modo sistemático, qualitativemente e quantitativamente a literatura. Para maiores informações consultar Hachicha and Ghorbel. (2012). Em outras palavras, essa análise consiste em um método observacional utilizado para avaliar sistematicamente o conteúdo registrado (Kolbe and Brunette, 1991). De modo geral, o procedimento para conduzir uma análise de conteúdo é baseado na definição das fontes e procedimentos de busca de artigos que serão analizados, assim como a definição das categorias para a classificação dos artigos selecionados. Nesta dissertação este procedimento está baseado em categorias para entender a história da aplicação das técnicas de *credit scoring*: ano da publicação, título da revista onde o artigo foi publicado, autor, técnica utilizada e medida de performance utilizada para avaliar o classificador obtido. Com esse objetivo, é preciso definir o critério de seleção dos artigos. Dois critéros de seleção são utilizados para selecionar os artigos relacionados com *credit scoring* usados nesse trabalho.

- O estudo é limitado aos trabalhos publicados nos seguintes banco de dados: Sciencedirect, Engineering Information, Reaxys e Scopus.

- A análise é restrita ao estudo de artigos publicados em Inglês, especialmente na área de *credit scoring* e com palavras chaves relacionadas como *machine learning*, *data mining*, *classification* ou *statistics*. Outras formas de publicações tais como artigos que não foram publicados, dissertação de mestrado e doutorado, livros, trabalhos de conferência, e outros

não são incluídos na revisão. Com os artigos que restaram, o horizonte da pesquisa cobre um período de 17 anos de: 1996 até 2012.

A Figura 1.1 exibe o procedimento utilizado na revisão bibliográfica sistemática.

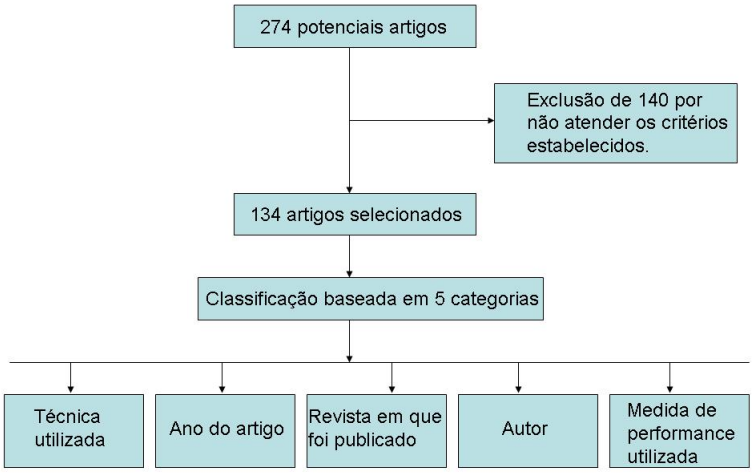


Figura 1.1: Procedimento utilizado na revisão bibliográfica sistemática.

Dos 274 artigos selecionados, que potencialmente abordavam *credit scoring*, 140 foram descartados devido ao fato de não atenderem o segundo critério. E portanto, como mostrado na figura 1.1, uma vez que o artigo foi selecionado coletamos as seguintes informações: autor, ano, revista, técnica utilizada e medida de performance utilizada.

1.1 Comentários Finais

Neste capítulo falamos de *credit scoring* de modo geral, algumas de suas aplicações e trabalhos publicados na área. Além disso, foi dito o que é uma revisão bibliográfica sistemática e também como foi realizada a seleção dos artigos.

Essa dissertação está organizada da seguinte maneira: Capítulo 2 fala sobre as principais técnicas de classificação que são utilizadas em *credit scoring*. Capítulo 3 contém as medidas de performance utilizadas em credit scoring. Capítulo 4 apresenta alguns resultados gerais tais como técnicas e medidas de performance mais comuns. O capítulo 5 faz uma comparação entre as três técnicas mais utilizadas, usando as medidas de performance coletadas dos artigos da revisão. O Capítulo 6 apresenta um estudo e simulação e no Capítulo 7 conclusão.

Capítulo 2

PRINCIPAIS TÉCNICAS DE CLASSIFICAÇÃO EM CREDIT SCORING

Nesta seção iremos apresentar as principais técnicas de classificação utilizadas em *credit scoring*. O objetivo é fornecer a ideia principal de cada técnica e subsídios para as próximas seções do texto.

2.1 Principais técnicas de classificação em *credit scoring*

Como foi abordado na seção anterior, existem várias técnicas de *credit scoring*. Nessa seção exibimos as principais técnicas aplicadas e exibimos, resumidamente, como utilizá-las.

Redes Neurais: E um sistema baseado na entrada de variáveis, também conhecido como variáveis explicativas, combinado com interações lineares ou não através de uma ou mais camadas, resultando em uma variável de saída, também conhecida como variável resposta (Ripley. 1996). Redes neurais foram criadas na tentativa de simular o cérebro humano, em que a informação é passada de um neurônio para outro através de estimulos elétricos. A estrutura da rede neural possui elementos que recebem um estimulo (variáveis de entrada), cria sinapses em diversos neurônios (ativação dos neurônios das outras camadas) e, resulta em uma resposta. As redes neurais diferem de acordo com suas estruturas básicas. Em geral, elas diferem no número de camadas ocultas e em suas funções de ativação.

Para tornar a ideia discutida acima mais clara, considere a Figura 2.1.

Na camada de entrada estão contidos os valores das covariáveis do indivíduo e, na camada oculta cada elemento $Z_i, i \in \{1, 2\}$ armazenará os seguintes valores,

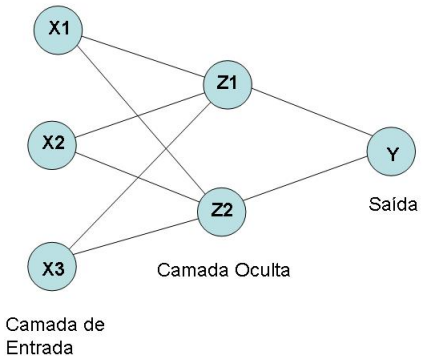


Figura 2.1: Exemplo de rede neural

$$Z_1 = \sigma(\alpha_{01} + \alpha_{11}X_1 + \alpha_{21}X_2),$$
$$Z_2 = \sigma(\alpha_{02} + \alpha_{12}X_1 + \alpha_{22}X_2)$$

Na camada de saída Y armazenará o valor obtido pela rede neural. Esse valor é calculado da seguinte forma.

$$Y = g(\beta_0 + \beta_1 X_1 + \beta_2 X_2)$$

Em que σ e g são funções. Os valores α_{ij} e β_i devem ser inicializados, esses valores são chamados de pesos e serão alterados a medida que a rede neural for "aprendendo"no processo de treinamento. O procedimento de treinamento da rede neural segue, basicamente, o seguinte esquema.

Para cada observação do banco de dados o erro é calculado $R = (y_i - \hat{y}_i)^2$, em que $\hat{y}_i$ é o valor estima pela rede neural. Os pesos serão atualizados da seguinte forma

$$\alpha_{k+1} = \alpha_k - \eta\frac{\partial R}{\partial \alpha}(k) \text{ e } \beta_{k+1} = \beta_k - \eta\frac{\partial R}{\partial \beta}(k) \tag{2.1}$$

Nas expressões apresentadas em 2.1, α_k e β_k, devem ser lidos como vetores e não como escalares. Assim como $\dfrac{\partial R}{\partial \alpha}$ é o vetor de derivadas de R em relação a cada um dos alfas. O procedimento descrito acima é conhecido como *gradient descendent* ou *backpropagation* e o valor η é conhecido como taxa de aprendizado (*learning rate*). Se lembrarmos que o gradiente de uma função aponta no sentido de crescimento da mesma, então esse algoritmo "caminha"no sentido oposto ao do crescimento do erro.

O algoritmo de treinamento é executado até que uma condição de parada seja satisfeita como, por exemplo, a soma dos quadrados dos erros seja menor que um erro pré-determinado ou, o algoritmo chegou no número máximo de iterações.

Vale salientar que existem outras estruturas mais sofisticadas de redes neurais. Maiores detalhes em Hastie, Tibshirani, Friedman. *The Elements of Statistical Learning.* 2 ed. Springer, 2013.

West (2000) mostra que *mixture-of-experts* e *radial basis function* devem ser consideradas no credit scoring. Leet et al(2002) propuseram uma modelagem hibrida em dois estágios para integrar análise de discriminante com redes neurais. Mais recentemente, diferentes tipos de redes neurais foram sugeridas para abordar o problema de *credit scoring*: Propabilistic neural network(Pang, 2005), partial logistic artificial neural network (Lisboa et al. 2009), artificial *metaplasticity neural network* (Marcano-Cedeno et al, 2011) e hybrid neural network (Chuang and Huang, 2011). Em alguns bancos de dados, as redes neurais conseguem produzir os melhores resultados em termos de acurácia (taxa de acertos) quando comparadas com outras técnicas, tais como análise de discriminante e regressão logística.

Support Vector Machine(SVM): Essa técnica é um método de classificação estatístico e foi introduzido por Vapnik(1998). Considere um conjunto de treinamento (x_i, y_i), com i = 1, ..., n em que x_i é o vetor de variáveis explicativas, y_i representa a categoria binária de interesse e, n denota a dimensão do vetor de entrada de dados. SVM tenta encontrar o hiper-plano ótimo. O hiper-plano ótimo pode ser escrito como segue:

$$\sum_{i=1}^{n} w_i x_i + b = 0,$$

em que $\mathbf{w} = w_1, ..., w_n$ é o vetor normal ao hiper-plano e, b é o intercepto. Considerando o hiper-plano separavél com respeito a $y \in \{-1, 1\}$ e com distância geométrica $\frac{2}{||\mathbf{w}||^2}$, o procedimeto maximiza essa distância, sujeito a restrição $y_i(\sum_{i=1}^{n} w_i x_i + b) \geq 1$. Geralmente, essa maximização é feita com multiplicadores de Lagrange e usando separação linear, polinomial, Gaussiana ou sigmoidal. Apenas recentemente SVM foi considerada como um modelo de *credit scoring* (Chen et al. 2009). Li et al. (2006); Gestel et al. (2006); Xiao and Fei. (2006); Yang. (2007); Chuang and Lin. (2009); Zhou et al. (2009, 2010); Feng et al. (2010); Hens and Tiwari. (2012); Ling et al. (2012) usaram SVM como técnica principal em seus trabalhos.

Regressão Linear: A regressão linear tem sido usada em aplicações de *credit scoring* mesmo quando a variável resposta é binária. A técnica atribui uma relação linear entre as características do cliente $X = \{X_1, ..., X_p\}$ e a variável resposta Y, da seguinte forma,

$$Y = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p + \epsilon,$$

em que ϵ é o erro aleatório e independente de X. A tradicional técnica de mínimos quadrados pode ser utilizada para estimar $\beta = \beta_0, ..., \beta_p$, sendo $\hat{\beta}$ o vetor estimado. Além disso, sendo Y uma variável binária, a esperança condicional $E(Y|X) = x'\beta$ pode ser usada para separar bons clientes e maus clientes. Uma vez que $-\infty < x'\beta < \infty$, a saída do modelo não pode ser interpretada como uma probabilidade. Hand e Kelly. (2002) construiram um modelo baseado em regressão linear (*superscore card model*). Karlis and Rahmouni(2007) propuseram um modelo de mixtura de Poisson para *credit scoring*. Outros autores usaram regressão linear ou suas generalizações (Hand and Kelly, 2002; Banasik et al, 2003; Karlis and Rahmouni, 2007; Efromovich, 2010).

Árvores de Decisão(Tree): Árvore de decisão é uma técnica utilizada para classificação, sendo um método muito comum no contexto de *credit scoring*. Essa técnica constrói um grafo conexo e sem circuitos, chamado de árvore, a partir dos dados fornecidos para treinamento, ao fim do processo de construção dessa estrutura a mesma é utilizada para classificação de novas instâncias.

Antes de detalhar um pouco mais essa técnica é importante saber que existem vários algoritmos para a construção de árvores de decisão como, por exemplo, ID3, CART, CHAID. Não iremos detalhar todos, apenas o ID3 para que possamos obter uma visão geral da técnica.

O algoritmo construirá uma estrutura similar como a apresentada na Figura 2.2

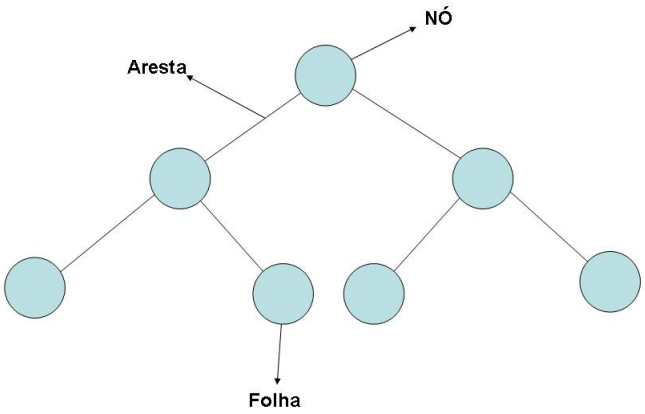


Figura 2.2: Exemplo genérico de árvore

Para facilitar vamos supor que as covariáveis são discretas então, cada nó representa uma covariável e as arestas são os possíveis valores dessa covariável. As folhas, no fim da árvore, representam as classes. Então para classificar uma nova instância a árvore testa os valores das covariáveis dessa instância e no final dos testes classifica em uma determinada classe fornecendo

assim uma classificação para essa instância. O próximo passo é saber como selecionar a covariável a ser ramificada.

Vamos definir uma medida chamada de entropia (Hastie, Tibshirani, Friedman, 2013), que mede o grau de impureza dos dados. Entropia é dada pela fórmula 2.2 em que S é o conjunto de dados, c é a quantidade de classes e p_i é a proporção da i-ésima classe.

$$E(S) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{2.2}$$

No contexto de *credit scoring* c = 2, pois temos duas classes apenas. Então, $E(S) = -p_0 \log_2(p_0) - p_1 \log_2(p_1)$. Agora suponha que tenhamos um conjunto de dados S totalmente puro, isto é, que possua apenas uma das classes, suponhamos que S contenha apenas indivíduos da classe 0, a entropia nesse caso assume o seguinte valor $E(S) = -p_0 \log_2(p_0) = -1 \log_2(1) = 0$. Note que não calculamos a segunda parcela do somatório 2.2, pois a classe 1 não esté presente em S.

Portando, no exemplo acima obtivemos um grau de impureza mínimo. Um resultado análogo é obtido se considerarmos um conjunto de dados S com apenas elementos da classe 1.

No contexto de duas classes, o conjunto de dados mais impuro possível seria quando metade dos elementos fossem da classe 0 e a outra metade da classe 1. Nesse caso, é fácil ver que *E(S)* = 1.

Uma outra medida que será utilizada para selecionar a covariável para ramificação é o ganho de informação. Dada uma covariável calculamos o ganho de informação obtido através da fórmula 2.3, em que g são os possíveis valores que X_i pode assumir.

$$G(S, X_i) = E(S) - \sum_{g \in X_i} \frac{|S_g|}{|S|} E(S_g) \tag{2.3}$$

O algoritmo é recursivo e funciona da seguinte forma: para cada covariável é calculado o ganho de informação e aquela com maior ganho é escolhida para ser ramificada. Uma vez ramificada uma covariável a mesma não é mais selecionada. As ramificações são feitas até que não seja mais possível ramificar, isto é, quando não houver mais covariáveis ou quando conjunto de dados possuir apenas uma única classe e então, um nó folha é criado. A seguir apresentamos o algoritmo extraído de Mitchell, Tom. *Machine learning*, McGraw-Hill, 1997.

Algoritmo 1: Algoritmo ID3.

Entrada: (Exemplos, atributos)

Saída: Saída do algoritmo é uma estrutura de árvore.

início

Criar o nó raiz da árvore;

if *todos os exemplos são positivos* **then**

retorne um único nó raiz com rótulo = +;

Pare;

if *todos os exemplos são negativos* **then**

retorne um único nó raiz com rótulo = -;

Pare;

if *covariáveis estão vazias ou nulas* **then**

retorne um único nó raiz com rótulo = classe mais frequente no banco de dados Exemplos;

Pare;

A = covariável do conjunto Exemplos que apresenta o maior ganho de informação;

Atribuir nó raiz = A;

repita

Adicionar uma aresta no nó A, correspondendo A = v_i;

Seja $Exemplos_{vi}$ o subconjunto de Exemplos tal que A=v_i;

if $Exemplos_{vi}$ *é vazio* **then**

Adicionar uma folha com rótulo = classe mais comum do nível anterior.;

else

ID3($Exemplos_{v_i}$, atributos - A);

end

até *Para cada possível valor,* $v_i \in A$;

fim

Além de vários algoritmos para a construção das árvores, existem vários métodos que podem ser utilizados para que o poder preditivo aumente entretando, esse métodos podem ser utilizados com outras técnicas e não apenas no caso de árvores de decisão. Alguns desses métodos são, *bagging* e *random forests* que fazem uso de técnicas de *bootstrap*. A ideia básica por trás desses métodos é utilizar várias amostras *bootstrap* para treinar várias vezes o método e combinar suas respostas para obter uma classificação mais precisa. Para maiores detalhes sobre árvores Breiman et al. (1984).

Regressão Logística: Esse modelo foi proposto por Berkson(1944), é um modelo muito utilizado não somente em *credit scoring* mas em qualquer tipo de problema de predição em que a variável resposta é binária. Esse modelo supõe que a variável resposta de cada cliente segue uma distribuição de bernoulli e que as covariáveis desse cliente são conhecidas. Então, o modelo logístico modela a probabilidade do evento de interesse através da expressão exibida em 2.4, em que $\mathbf{X_i}$ é o vetor contendo as informações do indivíduo. Na expressão 2.4 π_i representa a probabilidade de ocorrer o evento de interesse.

$$\pi_i = \frac{\exp(\mathbf{X_i}\beta)}{1 + \exp(\mathbf{X_i}\beta)} \tag{2.4}$$

Os coeficientes β_i são estimados pelo método da máxima verossimilhança e, a função de verossimilhança é obtida substituindo a expressão 2.4 na distribuição bernoulli. A função de verossimilhança é exibida em 2.5.

$$L(\beta) = \prod_{i=1}^{n} (\frac{\exp(\mathbf{X_i}\beta)}{1 + \exp(\mathbf{X_i}\beta)})^{y_i} (1 - \frac{\exp(\mathbf{X_i}\beta)}{1 + \exp(\mathbf{X_i}\beta)})^{1-y_i} \tag{2.5}$$

A função 2.5 é maximizada através de métodos numéricos. Alguns trabalhos que envolveram esse método são Li and Hand., 2002; Hand, 2005; Lee and Chen, 2005; Abdou et al, 2008; Yap et al, 2011; Pavlidis et al, 2012; Louzada et al, 2011.

Probito: A regressão probito é muito parecida com a logística, a diferença reside na função de ligação. Enquanto a regressão logística utiliza-se de uma função de ligação logística, a regressão probito utiliza a distribuição acumulada normal padrão, exibida em 2.6.

$$\pi_i = \phi(\eta_i) = \int_{-\infty}^{\eta_i} \frac{1}{\sqrt{2\pi}} \exp(-\frac{u^2}{2}) du \tag{2.6}$$

Em que $\eta_i = X_i\beta = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip} = X_i\beta$. No caso probito os coeficientes betas também são estimados através do método da máxima verossimilhança e a função de verossimilhança é construída de maneira análoga ao caso logístico.

Lógica Fuzzy(Fuzzy): A Lógica *fuzzy* foi introduzida por Zadeh (1965) como um sistema matemático que trata de modelagens com informações imprecisas na forma de termos linguísticos, fornecendo uma resposta aproximada baseada no conhecimento que pode ser impreciso, incompleto ou não totalmente confiável. Ao contrário da lógica binária, lógica *fuzzy* utiliza uma noção de semelhança para tratar com informações imprecisas. Um conjunto *fuzzy* é

unicamente determinado por uma função de semelhança, que pode ser triangular, trapezoidal, Gaussiana, polinomial ou sigmoidal. Hoffmann et al (2002) realizaram uma avaliação de dois classificadores *fuzzy* para *credit scoring*. Laha (2007) propos um método de construir modelos de *credit scoring* baseado em regras *fuzzy*. Lahsasna et al(2010) investigou o uso dos modelos Takagi-Sugeno (TS) e Mamdani para *credit scoring*.

Programação Genética(Genético): A programação genética é baseada em otimização matemática, como algoritmos heurísticos adaptativos, sua formulação foi inspirada pelo mecanismo de seleção natural e genética (Koza , 1992). Basicamente, o principal objetivo do algoritmo genético é criar uma população de respostas possíveis para um problema e então, submeter essa população ao processo de evolução aplicando operações genéticas tais como *crossover*, mutação e reprodução. O *crossover* é responsável pela troca de *strings* para gerar uma nova observação. Figura 2.3 mostra o processo de otimização do algoritmo genético. Ong et al. (2005) propuzeram um modelo de *credit scoring* genético e compararam esse modelo com técnicas tradicionais. Huang et al(2006) induziram programação genética em dois estágios. Muitos outros autores investigaram modelos genéticos em aplicações de *credit scoring* (Chen and Huang 2003; Mues et al 2004; Abdou, 2009; Won et al, 2012).

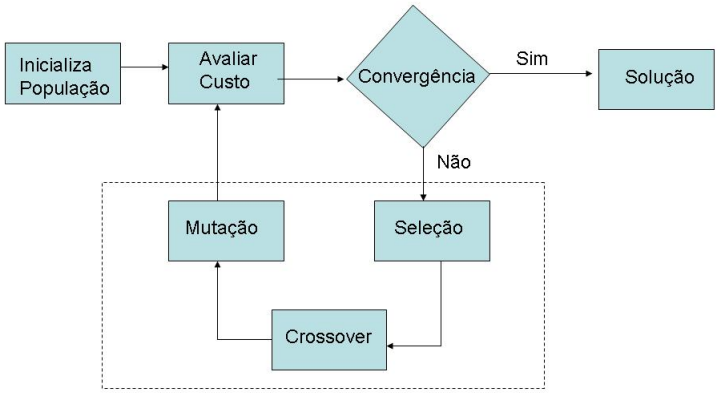


Figura 2.3: Esquema de algoritmo genético, adaptado de Abdoun and Abouchabaka (2011).

Análise de discriminante(DA): Introduzida por Fisher em (1936) essa técnica é bastante popular em *credit scoring*. Existem duas versões dessa técnica que são, análise de discriminante linear(LDA) e análise de discriminante quadrática (QDA).

A diferença é que no segundo caso, QDA, as matrizes de variância e covariância são distintas. Vamos explicar o funcionamento da técnica. Sejam $P(Y = 1|\mathbf{x})$ e $P(Y = 0|\mathbf{x})$ a probabilidade do cliente ser default e não ser default, respectivamente.

Usando o Teorema de Bayes podemos escrever as probabilidades da seguinte forma:

$$P(Y = k|\mathbf{x}) = \frac{P(\mathbf{x}|Y = k)P(Y = k)}{P(\mathbf{x})},\ k \in \{0, 1\} \tag{2.7}$$

Supondo que $\mathbf{x}$ seja um vetor de dimensão p, vamos admitir que $\mathbf{x}$ tenha uma distribuição normal p-variada com os vetores de médias μ_1 e μ_0, isto é, μ_1 vetor de médias para os default e μ_0 para os não default e, Σ é a matriz $p \times p$ de variância e covariância comum para ambos os tipos de clientes. Sejam $\hat{\mu_k}$ e $\hat{\Sigma}$ as estimativas de máxima verossimilhança dos parâmetros, em que $k \in \{0, 1\}$.

A expressão da normal p-variada é dada por:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\Sigma|^{1/2}} \exp(-\frac{1}{2}(\mathbf{x} - \mu)^t\Sigma^{-1}(\mathbf{x} - \mu)) \tag{2.8}$$

Substituindo 2.8 em 2.7, tomando $P(Y = k) = \pi_k$, aplicando o log na probabilidade, realizando algumas manipulações algébricas e desconsiderando os termos que não dependem de k, obtemos a seguinte expressão.

$$\log(P(Y = k|\mathbf{x})) = \mathbf{x}^t\hat{\Sigma}^{-1}\hat{\mu_k} - \frac{1}{2}\hat{\mu_k}^t\hat{\Sigma}^{-1}\hat{\mu}_k + \log(\hat{\pi}_k) \tag{2.9}$$

Note que estamos trabalhando com o log na base e, também utilizamos o seguinte argumento. $\log(P(Y = 1|\mathbf{x})) > \log(P(Y = 0|\mathbf{x})) \Rightarrow P(Y = 1|\mathbf{x}) > P(Y = 0|\mathbf{x})$, pois a função logaritmica é monótona e crescente. Note também que cometemos um abuso de notação ao utilizar o sinal de igualdade em 2.9, pois na verdade na expressão 2.9 desconsideramos os termos que não dependem da classe k. Para obtermos a expressão no caso QDA o procedimento é análogo, porém as matrizes de variância e covariância não serão iguas.

Portanto, dadas as covariáveis do indivíduo, ou seja, $\mathbf{x}$, atribuímos esse indivíduo a classe que possuir o maior valor em 2.9. Alguns autores que utilizaram essa técnica são West, 2000; Gestel et al, 2006; Akkoc, 2012, Yang, 2007; Falangis and Glen, 2010.

K-Nearest Neighbor: A técnica nearest neighbor utiliza alguma métrica de distância para classificar uma observação. Por exemplo, supondo que possuímos uma observação (x_i) que deseja-se classificar. Então, selecionando as k observações do banco de dados mais próximas, utilizando alguma métrica de distância como, por exemplo, a distância eucludiana e, calculando

a média das respostas dessas k observações obtém-se uma estimativa para a classificação da observação x_i.

Apenas para título de conhecimento, é interessante citar as técnicas de clusters que visam dividir os dados em grupos homogêneos. Essas técnicas também utilizam métricas de distâncias para alcançar esse objetivo.

Análise de Sobrevivência(Survival): Uma outra abordagem para trabalhar com dados de crédito é através da análise de sobrevivência na tentativa de prever o tempo até o *default*. Existem duas formas bastante comuns de trabalhar com variáveis explicativas em sobrevivência, sendo elas, tempo de vida acelerado e modelo de riscos proporcionais. No tempo acelerado, o modelo assume a seguinte forma

$$h(t) = \exp(\eta)h_0(t\exp(\eta)) \text{ e } S(t) = S_0(t\exp(\eta))$$

Em que, $\eta = \beta_0 + \beta_1 x_{i1} + ... + \beta_p x_{ip}$, $h_0(t)$ e $S_0(t)$ são a função de sobrevivência e de risco, respectivamente. Note que $h_0(t)$ e $S_0(t)$ podem ser obtidas usando qualquer densidade de probabilidade; sendo exponencial e Weibull as escolhas mais comuns.

No caso de riscos proporcionais, o modelo assume a seguinte forma:

$$h(t) = \exp(\eta)h_0(t) \text{ e } S(t) = S_0(t)^{\exp(\eta)}$$

Da mesma forma como dito acima, as funções $S_0(t)$ e $h_0(t)$ são obtidas a partir de uma densidade de probabilidade e o método de máxima verossimilhança é utilizado para estimar os coeficientes β_i. Entretanto, Cox(1972) mostrou que os parâmetros podem ser estimados sem qualquer conhecimento de $h_0(t)$ (Stepanova, Thomas, 2000).

Redes Bayesianas: Uma rede bayesiana é um grafo direcionado e sem ciclos. Em que cada vértice representa uma variável aleatória e cada aresta representa a dependência entre duas variáveis. Para exemplificar a ideia, suponha que X_1, X_2, X_3 e X_4 são v.a's de modo que, $X_1 \in \{0,1\}$, $X_2 \in \{0,1\}$, $X_3 \in \{0,1\}$ e $X_4 \in \{0,1,2\}$. A Figura 2.4 exibe um exemplo desse tipo de rede.

As variáveis X_1 e X_2 são independentes, a variável X_3 apresenta uma dependência de X_1 e X_2 e por isso sua distribuição de probabilidade é calculada condicionando em X_1 e X_2 e, a distribuição de X_4 é calculada condicionando em X_3. Desta forma, a Figura 2.4 representa a distribuição conjunta das variáveis, isto é, $P(X_1, X_2, X_3, X_4) = P(X_1)P(X_2)P(X_3|X_1, X_2)P(X_4|X_3)$.

Giudici (2001) apresenta um grafo para extrair informações das associações de variáveis em aplicações de *credit scoring*. Gemela (2001) aplicou redes bayesianas em uma base de dados

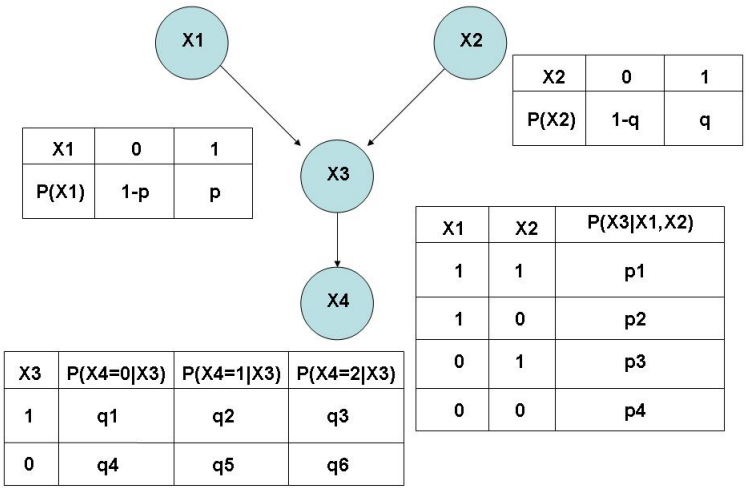


Figura 2.4: Exemplo de uma rede bayesiana.

de crédito. Outros autores que investigaram tais redes são Zhu et al (2002); Antonakis and Sfakianakis (2009); Wu (2011).

Como caso particular temos o classificador naive Bayes, que baseado no teorema de Bayes retorna a probabilidade de uma observação pertencer a uma determinada classe c_i dado um conjunto de variáveis explicativas.

$$p(c_i|x_1, ..., x_n) = \frac{p(c_i)p(x_1, ..., x_n|c_i)}{p(x_1, ..., x_n)}$$

Nós podemos trabalhar somente com o numerador porque o denominador, $p(x_1, ..., x_n)$, é constante. Rescrevendo obtemos:

$$p(c_i)p(x_1, ..., x_n|c_i) = p(c_i, x_1, ..., x_n) = p(c_i)p(x_1|c_i)p(x_2|c_i, x_1)...p(x_n|c_i, x_1, ..., x_{n-1})$$

Supondo x_i e x_j são independentes, $i \neq j$,

$$p(c_i)p(x_1, ..., x_n|c_i) = p(c_i)p(x_1|c_i)p(x_2|c_i)...p(x_n|c_i) = p(c_i)\prod_{j=1}^{n} p(x_j|c_i)$$

Dadas as variáveis explicativas, é necessário encontrar a classe c_i tal que a probabilidade $p(c_i|x_1, ..., x_n)$ é máxima.

Escrevendo matematicamente,

$$c_i = arg_{c_i \in C}\ max\{p(c_i)\prod_{j=1}^{n} p(x_j|c_i)\}$$

Os autores (Shinha and Zhao 2008) trabalharam com algumas técnicas dentre elas naive Bayes e com a incorporação do conhecimento do especialista no modelo.

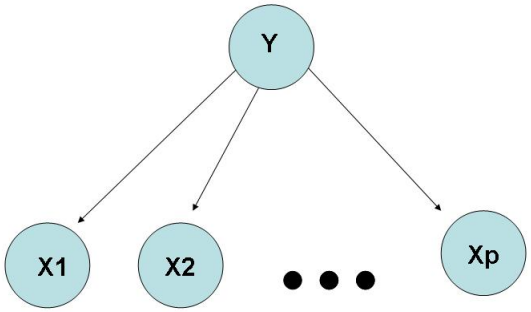


Figura 2.5: Rede no caso do naive Bayes

No caso do naive Bayes temos uma variável $Y \in \{0, 1\}$ representando *default* ou não *default* e as covariáveis $X_1, ..., X_p$ que são independentes (suposição do modelo), então a Figura 2.5 apresenta o esquema gráfico de como seria a rede para o caso do naive Bayes.

Programação matemática (Mat. Prog) Essa técnica utiliza métodos de otimização para construir um modelo de *credit scoring* (Thomas. et al, 2002), e existem várias formas de se modelar um dado problema, a seguir mostraremos uma delas. Suponha que dispomos de uma amostra $(y_i, x_{i1}, ..., x_{ip})$, i=1...n. Sem perda de generalidade, considere que os primeiros n_G são bons clientes e, $n_G + 1, ..., n$ são *default*. Precisamos encontrar um vetor $(\beta_1, ..., \beta_p)$ tal que, dado um ponto de corte λ e para o i-ésimo cliente, supondo que seja um cliente bom, $\beta_1 x_{i1} + ... + \beta x_{ip} \geq \lambda$ e no caso em que o cliente é mau, $\beta_1 x_{i1} + ... + \beta x_{ip} < \lambda$. Adicionando as variáveis $a_i \geq 0$ obtemos, $\beta_1 x_{i1} + ... + \beta x_{ip} \geq \lambda - a_i$ ou $\beta_1 x_{i1} + ... + \beta x_{ip} < \lambda + a_i$. Uma possível modelagem para o problema é dada por (Thomas. et al, 2002),

$$
\begin{gathered}
Min : a_1 + a_2 ... + a_n \\
\text{s.a: } \beta_1 x_{i1} + ... + \beta_p x_{ip} \geq \lambda - a_i,\ 1 \leq i \leq n_G \\
\beta_1 x_{i1} + ... + \beta_p x_{ip} < \lambda + a_i,\ n_G + 1 \leq i \leq n_B + n_G \\
a_i \geq 0, i = 1...n
\end{gathered}
$$

Uma vez estimados os betas é possível fornecer um *score* para um indivíduo qualquer e, com esse *score* decidir em inadiplente ou não.

Rough Sets: Com o avanço da tecnologia houve também um aumento no volume de dados, e com grandes bases de dados é preciso lidar com informações muitas vezes imprecisas ou que apresentam ruídos, é necessário encontrar os atributos que realemte são relevantes. Desenvolvida por Pawlak(1982), *rough sets* faz uso da teoria dos conjuntos e conceitos como relações

de equivalência e classes de equivalência para analisar os dados. Os dados em uma mesma classe são indiscerníveis. Além disso, essa técnica pode ser usada em problemas de classificação e para descobrir atributos relevantes, autores como Ong C. S, Huang. J. J, Tzeng. G. H, 2005 utilizaram esse método no problema de *credit scoring*.

Outras técnicas: Algumas vezes uma abordagem diferente é usada para tratar o problema de *credit scoring*. Por exemplo, Karlis. D, Rahmouni. M, (2007), usaram um modelo de mistura de Poisson; Abdou. H, (2009), trabalhou com weight of evidence; Chang. S. Y, Yeh, T. Y, (2011) usaram artificial immune network esse método é baseado no sistema de defesa do corpo humano; Martens *et al.* (2008) usaram um método baseado em colônias de formigas; Yu. J. L, Li. H, (2011) usaram *case based reasoning* (CBR); Luo. S. T, Cheng. B. W, Hsieh. C. H, (2009), usaram um algoritmo novo, *clustering-launched classification* (CLC), criado por (Chen, Lin, Chiu, Lin, & Chen, 2006); Giudici. P, (2001) usou cadeias de markov; Hand. D. J, Li. H. G, 2002 trabalharam com um método de classificação indireta; Zhu. H, Beling. P. A, Overstreet. G. A, (2002) usaram técnicas baseadas em métodos bayesianos e uma modelagem meta-Gaussiana.

2.2 Comentários Finais

Como foi visto, existem várias opções de métodos para o contexto de *credit scoring*, isto é, é possível empregar técnicas bastante variadas. Veremos mais adiante que, entre as mais conhecidas estão a regressão logística, árvores de decisão e redes neurais e também, algumas técnicas novas têm surgido como, por exemplo, SVM. Portanto, nessa seção foram exibidas as características básicas de cada técnica utilizada em *credit scoring* com o intuito de fazer apenas uma breve introdução. No próximo capítulo abordaremos tópicos como validação cruzada, e algumas das medidas preditivas como, por exemplo, KS, AUC, GINI entre outras.

Capítulo 3

PRINCIPAIS MÉTODOS DE AVALIAÇÃO DA CAPACIDADE PREDITIVA

Uma vez escolhido e ajustado o modelo é necessário checar a capacidade preditiva do mesmo. Existe uma grande quantidade de medidas para realizar tal tarefa. Aqui vamos exibir e explicar o uso das medidas mais comumente utilizadas. Em geral, os autores utilizam mais de uma medida. Antes de abordarmos tais medidas, falaremos sobre algumas técnicas que são utilizadas em conjunto com as medidas de performance, essas técnicas são conhecidas como processos de validação. Portanto, o objetivo do capítulo é apresentar um conjunto de técnicas que são utilizados para aferir o poder preditivo do modelo empregado.

Na seção 2.1 apresentaremos algumas formas de divisão do conjunto de dados que são muito utilizadas na prática. O conjunto de dados é dividido em treinamento e teste. Na seção 2.2 serão exibidos as medidas de performance.

3.1 Validação Cruzada

Validação cruzada é uma técnica utilizada no processo de avaliação da capacidade preditiva do modelo. Abaixo são exibidas as metodologias mais utilizadas.

Amostra holdout: Esse método é realizado particionando o conjunto de dados aleatoriamente em dois sub conjuntos. O primeiro conjunto é chamado de conjunto de treinamento e utilizado na fase de estimação dos parâmetros do modelo, o segundo é o conjunto de teste e, é utilizado para avaliar a capacidade preditiva do modelo. Se o modelo obtém uma boa performance no conjunto de teste isso indica que o modelo ajustou-se adequadamente aos dados e não ocorreu o *overfitting* no conjunto de treinamento. Uma possível escolha é utilizar 2/3 dos dados

para treino e 1/3 para teste.

K-fold: Esse método é uma generalização do método holdout, em que o banco de dados é particionado aleatoriamente em K sub conjuntos. Cada sub conjunto é usado como conjunto de teste para o modelo considerando os outros K-1 conjuntos para treinamento. Nessa abordagem, todo o banco de dados é usado para treinamento e para teste do modelo. Tipicamente, o valor k=10 é usado na literatura (Mitchell, 1997). Essa técnica é útil para verificar se uma pequena mudança na amostra provoca uma grande diferença nos parâmetros, isto é, essa técnica é utilizada para verificar a estabilidade do modelo. Além disso, é possível obter uma amostra de valores de alguma medida de performance que pode ser utilizada para comparação entre modelos. A principal desvantagem é o aumento no tempo computacional e, no caso de uma base de dados pequena pode ser complicado fazer essa divisão.

Leave One Out: Esse método é um caso particular do K-fold em que, K é igual ao tamanho da base de dados. Cada caso é usado para testar o modelo e os outros n-1 são utilizados no ajuste(treino) do modelo. Nessa abordagem o todo o banco de dados é usado tanto para treino e teste do modelo. É importante notar também que essa técnica, assim como a anterior, eleva o custo computacional.

3.2 Medidas de Performance

Curva ROC: A curva ROC (Reciever Operating Characteristic) pode ser geometricamente definida como um gráfico para visualizar a performance de uma técnica de classificação binária Zweig e Campbell (1993). A curva ROC é obtida plotanto 1-especificidade no eixo abscissa (x) e, sensibilidade na ordenada (y). Assim, quanto mais a curva se afasta da bissetriz do primeiro quadrante melhor é modelo. A Figura 3.1 exibe um exemplo da curva ROC.

Existem duas medidas de performance bastantes comuns que são baseadas na curva ROC, as medidas são a área sob a curva (AUC) e GINI. Em geral, o modelo que produz a curva mais distante da diagonal é considerado como o método mais adequado. Então, de acordo com o AUC, o modelo que possuir um valor maior para essa medida será considerado o melhor modelo. O coeficiente de GINI está muito relacionado com o AUC e algumas vezes é usado em seu lugar. O coeficiente de GINI pode ser calculado usando a seguinte fórmula GINI + 1 = 2AUC (Hand. D, Till. R, 2001), alguns autores Finlay. S, 2008; Bijak. L, Thomas, 2012; Finalay, 2009 usaram o GINI.

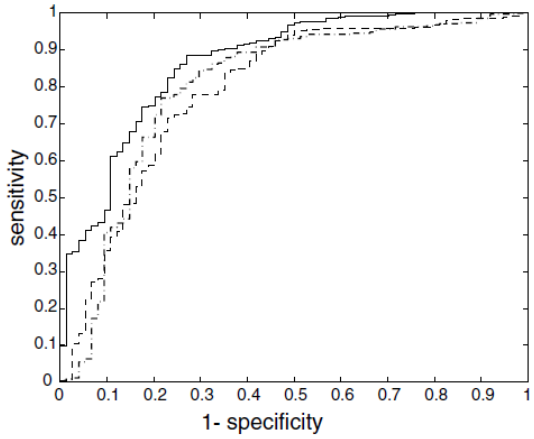


Figura 3.1: Curva ROC usada por Gestel et al. (2006) para comparar *support vector machine* (linha sólida), regressão logística (ponto-traço) e *linear discriminant analysis* (traços).

Métricas baseadas na matriz de confusão: O objetivo é comparar a capacidade preditiva dos modelos através das respostas fornecidas pelos mesmos e a verdadeira resposta na base de dados. Um erro de classificação ocorre quando o modelo falha em alocar corretamente um indivíduo em sua categoria correta. Um procedimento tradicional é construir a matriz de confusão, como mostrado em 3.1 em que, M é o resultado do modelo e, D é o valor real na base de dados. TP é o número de verdadeiros positivos (*True Positive*), FP é o número de falsos positivos (*False Positive*), FN é o numero de falsos negativos (*False Negative*) e TN é o numero de verdadeiros negativos (*True Negative*). Naturalmente, $TP+FP+FN+TN$=N, em que N é o número de observações.

Tabela 3.1: Matriz de confusão.

		M	
		$\{1\}$	$\{0\}$
D	$\{1\}$	TP	FN
	$\{0\}$	FP	TN

Através da matriz de confusão, algumas medidas são empregadas para avaliar a performance do modelo.

Acurácia (ACC): É uma medida muito comum na literatura, ACC é a razão de todas as predições corretas do modelo dentro de cada classe $\{0\}$ ou $\{1\}$. A expressão é ACC $= \frac{TP+TN}{TP+TN+FN+FP}$. ACC é uma medida de fácil interpretação e, uma vez calculada a matriz de confusão, muito fácil de ser obtida. Quanto maior o valor de ACC melhor é o modelo, sendo que essa medida varia em [0, 1]. Valores comuns na literatura estão em torno de 70% a 90% e, algumas vezes mais de 90% (Baesens *et al.*, 2003; Tian-Shyug Lee *et al.*, 2002; Lin Feng *et al.*, 2010). Entretando, a ACC assume os mesmos custos para os erros, FP e FN, e essa suposição pode

ser muito severa em problemas reais, por exemplo, no caso de uma amostra muito desbalanceada.

Taxa de erro (Error Rate - ER): A taxa de erro pode ser calculada usando a matriz de confusão ou usando o ACC, essa medida pode ser considerada equivalente à medida anterior (ACC). A expressão é dada por $ER = \frac{FP+FN}{TP+TN+FN+FP} = 1-ACC$. Essa medida também possui uma interpretação muito intuitiva e, é fácil de ser calculada porém, possui o mesmo problema da medida ACC quando o os erros possuem custos distintos. Alguns dos autores que utilizaram essa medida foram, Rudy Setiono et al., 2008; Yingxu Yang. 2007.

Sensibilidade (SEN): Essa medida é muito útil e, também é derivada da matriz de confusão. Essa medida mostra a capacidade do modelo em detectar o evento dado que o evento realmente ocorreu. A sensibilidade é dada por $SEN = \frac{TP}{TP+FN}$. Quando o modelo possui uma alta sensibilidade isso é indicativo de que esse modelo é bom em detectar o evento de interesse. Sensibilidade também é conhecida como *Recall* ou *True Positive Rate*.

Especificidade (SPE): Também conhecida como *True Negative Rate*, é similar a sensibilidade. Essa medida é obtida pela razão das observações corretamente classificadas como não evento $\{0\}$ dentre todas as observações que realmente pertencem a classe $\{0\}$. Sua expressão é dada por $SPE = \frac{TN}{TN+FP}$. Essa medida indica o quão bom é o modelo para detectar não evento dado que o evento realmente não ocorreu.

Taxa de Falso Negativo (FNR): Também conhecido como erro tipo I é a fração de casos $\{1\}$ que são classificados de maneira equivocada na classe $\{0\}$. A expressão é dada por $FNR = \frac{FN}{TP+FN} = 1 - SEN$. Em outras palavras, o erro tipo I fornece uma estimativa da probabilidade do modelo não detectar o evento dado que o evento ocorreu. Como essa medida foi dada em função da sensibilidade, ambas foram colocadas no mesmo grupo da sensibilidade.

Taxa de Falso Positivo (FPR): Também conhecido como erro tipo II, é a proporção de casos $\{0\}$ classificados incorretamente na classe $\{1\}$. A expressão da medida é dada por $FPR = \frac{FP}{TN+FP} = 1 - SPE$. Em outras palavras, o erro tipo II estima a probabilidade do modelo detectar o evento dado que o evento não ocorreu. Assim como no caso anterior, essa medida foi agrupada juntamente com a especificidade.

Taxa de maus entre os aceitos (BRA): BRA é a abreviação de *Bad Rate Amongst Accepted*, como foi dito os tipos de erros podem apresentar custos diferentes e na literatura o erro cometido ao classificar um *default* como não *default* é pior do que classificar errôneamente um

não *default* como *default*. Então, a medida BRA é uma das medidas que podem ser utilizadas em *credit scoring*. Em modelo de *credit scoring* é desejado que o valor do BRA seja o menor possível, autores que fizeram uso dessa medida são Hand. (2005); Hand, Kelly. M. G, (2002); Antonakis. A. C, Sfakianakis. M. E, (2009). É possível escrever a expressão do BRA em termos dos elementos da matriz de confusão, primeiro vamos definir a classe {1} como sendo a *default* e {0} como não *default* então, $BRA = \frac{FN}{TN+FN}$.

Verdadeiro Predito Positivo(TPP): *True Predicted Positive* é uma outra opção para avaliar a performance do modelo, sua expressão é dada por $TPP = \frac{TP}{TP+FP}$ é a fração dos indivíduos classificados corretamente na classe {1} dentre todos aqueles que foram classificados como pertencentes a essa classe. Essa medida é também conhecida como precisão.

Verdadeiro Predito Negativo (TPN): essa medida é definida da seguinte forma $TPN = \frac{TN}{TN+FN}$ e é a fração dos indivíduos que foram classificados corretamente na classe {0} dentre todos aqueles que foram classificados como pertencentes a classe {0}.

Misclassification cost (Miscla. Cost): Como é comum os erros terem custos diferentes, então é interessante encontrar uma expressão para avaliar o modelo e que considere essa questão. Seja C_{12} o custo de classificar um *default* como não *default* e, C_{21} o custo de classificar como *default* um não *default*. Os custos de cada tipo de erro sugerido por Hoffmann são, $C_{12} = 5$ e $C_{21} = 1$, a expressão 3.1 fornece uma maneira de medir a performance do modelo considerando os custos (West 2000).

$$custo = C_{12}\pi_2(n_2/N_2) + C_{21}\pi_1(n_1/N_1) \tag{3.1}$$

Em que π_1 é a probabilidade do cliente ser bom e π_2 é a probabilidade do cliente ser mau. Essas probabilidades podem ser estimadas a partir da base de dados. Além disso, $\frac{n_2}{N_2}$ é a probabilidade do modelo classificar como bom um indivíduo que é mau e $\frac{n_1}{N_1}$ é a probabilidade do modelo classificar como mau um indivíduo que é bom. Note que $\frac{n_1}{N_1}$ e $\frac{n_2}{N_2}$ podem ser obtidas a partir da matriz de confusão.

F-Measure: Essa medida faz uso de algumas medidas descritas anteriormente. A expressão é dada por $F = \frac{2}{1/TPP+1/SEN}$. Essa medida foi usada por Wang. J et al, (2001) para comparar modelos entretanto, não é muito comum observar seu uso. Interessante notar que essa medida faz o uso combinado da sensibilidade e verdadeiro predito positivo. A medida F terá seu valor aumentado quando $1/SEN$ e $1/TPP$ possuirem valores baixos, ou seja, TPP e sensibilidade

devem possuir valores altos,

$0 \leq F \leq 2$.

AIC, BIC, log-likelihood: Essas medidas são aplicadas no caso em que técnicas estatísticas são utilizadas. Por exemplo, suponha que dispomos de uma base de dados e é necessário escolher entre dois modelos, exponencial ou Weibull. É sabido que a exponencial é um caso particular da Weibull, então a Weibull tem um poder maior para se ajustar aos dados porém, a exponencial possui uma interpretação mais fácil. Nesse caso é possível utilizar essas medidas para avaliar qual modelo é mais adequado. Essas medidas levam em conta a quantidade de parâmetros no modelo (modelos encaixados) e o modelo com mais parâmetros é penalizado. Entretanto, essas medidas não podem ser utilizadas para comparar modelos de natureza distintas, por exemplo, não é possível comparar um modelo de regressão logística com um modelo de árvore de decisão.

Komolgorov-Smirnov (KS): Originalmente o KS é usado para testar se uma dada amostra é oriunda de uma determinada densidade (distribuição) de probabilidade, isso é feito calculando a máxima diferença entre a distribuição empírica e a teórica. No contexto do *credit scoring*, o KS é obtido calculando a máxima diferença entre as proporções acumuladas dos *default* e os não *default* porém, o KS é mais comum na indústria do que em pesquisas acadêmicas. Em geral, um modelo com o valor KS mais elevado é o melhor modelo. A Figura 3.2 apresenta uma tabela com as proporções acumuladas de clientes bons e maus para cada valor da faixa do *score*, variando de 1 até 10, e o valor máximo, em módulo, da diferença é 30,56 então, o valor do KS é 30,56.

Score category	Total number	Cum. %	Number of goods	Number of bads	Cum. % Goods	Cum. % Bads	The difference %	Cum. odds
10	200,000	10	192,000	8,000	10.7	4.0	6.67	24.00
9	200,000	20	191,000	9,000	21.3	8.5	12.78	22.53
8	200,000	30	190,000	10,000	31.8	13.5	18.33	21.22
7	200,000	40	188,000	12,000	42.3	19.5	22.78	19.51
6	200,000	50	186,000	14,000	52.6	26.5	26.11	17.87
5	200,000	60	185,000	15,000	62.9	34.0	28.89	16.65
4	200,000	70	183,000	17,000	73.1	42.5	30.56	15.47
3	200,000	80	179,000	21,000	83.0	53.0	30.00	14.09
2	200,000	90	171,000	29,000	92.5	67.5	25.00	12.33
1	200,000	100	135,000	65,000	100.0	100.0	0.00	9.00
Total	2,000,000		1,800,000	200,000			30.56	9.00

Figura 3.2: Figura exibida por Dryver. A, Sukkasem, 2009 para cálculo do KS.

Técnicas auxíliares: É bastante comum a utilização de técnicas auxíliares como o McNemar's test, teste de Wilcoxon, Friedman-test para verificar se há diferença estatisticamente significativa entre as medidas. Por exemplo, consideremos que um processo de validação cruzada tenha sido empregado resultando em 10 valores de ACC, então podemos aplicar os testes citados

para verificar se um modelo é melhor que outro. Chen, Li. (2010); Chang, Yeh. (2011); Won, Kim, Bae. (2012); Marqués, et al. (2012); S. Vukovic. et al. (2012); West (2000) utilizaram esse tipo de procedimento. Outra possibilidade é utilizar um teste paramétrico como, por exemplo, o teste t.

Kaplan-Meier: O estimador Kaplan-Meier estima a curva de sobrevivência e é bastante comum comparar a curva de sobrevivência empírica estimada pelo Kaplan-Meier com a curva obtida pelo modelo ajustado. A Figura 3.3 exibe um exemplo dessa situação.

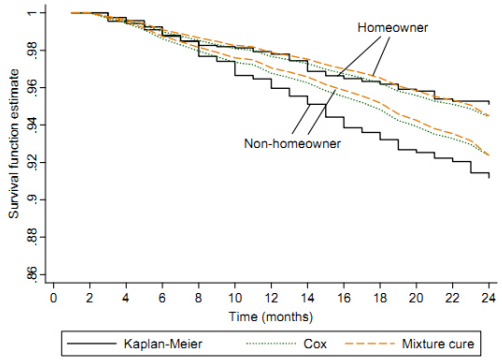


Figura 3.3: Imagem de Tong. E, et al. 2012

Após ajustado um modelo de sobrevivência, espera-se que a curva de sobrevivência obtida pelo modelo esteja razoavelmente próxima da curva Kaplan-Meier.

As técnicas auxíliares e o Kaplan-Meier não foram consideradas como medidas de performance, citamos essas técnicas aqui apenas a título de conhecimento.

3.3 Comentários Finais

Essa seção exibiu as principais técnicas e medidas utilizadas para avaliar o poder preditivo e também abordou a validação cruzada que é utilizada juntamente com essas medidas. A validação cruzada é útil para detectar um fenômeno conhecido como *overfitting* pois, o modelo tende a ter uma performance melhor quando avaliado nos dados de treinamento. Isso ocorre porque o modelo incorpora particularidades dos dados. Então, é comum dividir o conjunto de dados em amostras treinamento e teste. Na próxima seção exibiremos alguns resultados obtidos.

Capítulo 4

RESULTADOS GERAIS

Nesse capítulo apresentamos alguns resultados gerais. Mostraremos informações tais como, as técnicas mais utilizadas nos artigos, medidas de performance mais utilizadas e as revistas científicas. O intuito desta seção é obter uma visão geral do cenário de alguns anos atrás até o cenário atual com relação aos métodos utilizados.

4.1 Descrição dos artigos revisados

Na Figura 4.1 podemos ver a quantidade de artigos publicados em cada ano de 1996 até 2012. Devido ao grande interesse em técnicas de *credit scoring*, fica evidente um aumento na quantidade de artigos publicados a partir de 2000. Além disso, podemos ver um pico próximo de 2010.

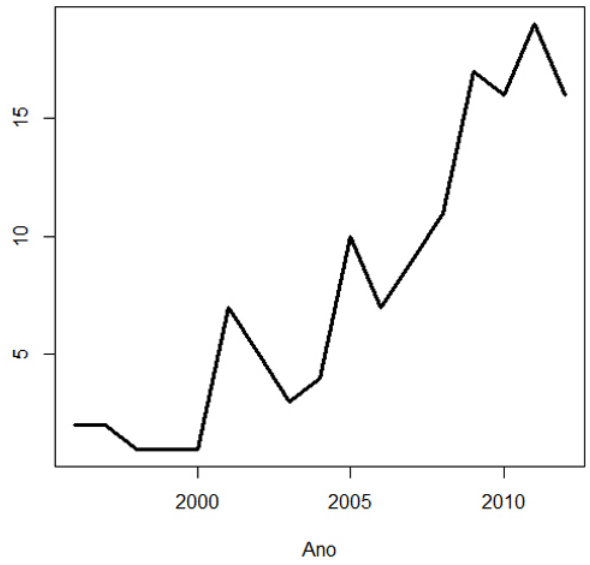


Figura 4.1: Quantidade de artigos publicados em cada ano.

Os artigos foram publicados em várias revistas e as frequências (relativas e absolutas) são

mostradas na Tablela 4.1. A maioria dos artigos estão relacionados com ciência da computação, ciência de decisão, engenharia e matemática. Como pode ser visto na Tabela 4.1, o maior número de artigos foram publicados por "Expert Systems with Apllication"e "Journal of the Operational Research Society"com porcentagens de 29,10 e 8,96 respectivamente.

Tabela 4.1: Distribuição dos artigos revisados de acordo com o nome da revista.

Journal	Percentage	Total
Expert Systems with applications	29,10	39
Journal of the Operational Research Society	8,96	12
European Journal of Operational Research	7,46	10
Applied Stochastic Models in Business and Industry	2,99	4
Knowledge-Based Systems	2,24	3
IMA Journal of Management Mathematics	2,24	3
Computational Statistics and Data Analysis	1,49	2
International Journal of Forecasting	1,49	2
Journal of Applied Statistics	1,49	2
IEEE Transactions on Neural Networks	1,49	2
Others†	41,04	55
Total	100	134

Na Figura 4.2 são mostradas as técnicas mais popularmente usadas no problema de *credit scoring*. A regressão logística é a mais comum, seguida das redes neurais. A abordagem por rede neural tem recebido muita atenção devido à sua capacidade de ajuste quando as relações não são lineares. Além disso, muitos trabalhos obtém uma ACC bastante elevada com essa técnica, porém ela é muito criticada pela sua dificuldade de interpretação, isto é, é complicado saber qual característica tem maior impacto na variável resposta. O modelo logístico é muito utilizado pela fácil interpretação. Entretanto, esse modelo faz a suposição de que as variáveis explicativas contribuem linearmente no valor do η_i.

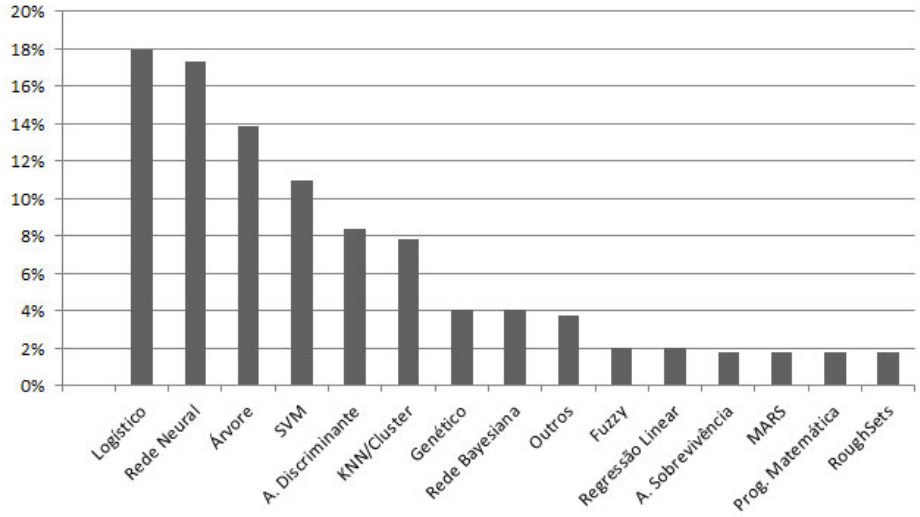


Figura 4.2: técnicas

Notamos que é muito comum a aplicação de técnicas não estatísticas como, por exemplo, redes neurais e SVM. Isso mostra a multidisciplinaridade da área.

Na Figura 4.3 podemos ver as medidas de performance mais comuns e, na Figura 4.4, observamos as técnicas com a distribuição das medidas de performance mais utilizadas. Como podemos ver a medida ACC é muito utilizada mesmo com a restrição com relação aos custos de cada erro. Sua alta popularidade deve-se a fácil interpretação do valor obtido. A sensibilidade e especificidade também são muito comuns pois, medem a capacidade preditiva do modelo dentro de cada classe. Algumas vezes os autores tentam reduzir um determinado tipo de erro através de combinações de técnicas. Chuang. C. L, Lin. R. H, (2009) trabalharam para reduzir a quantidade de clientes bons que são rejeitados pelo modelo.

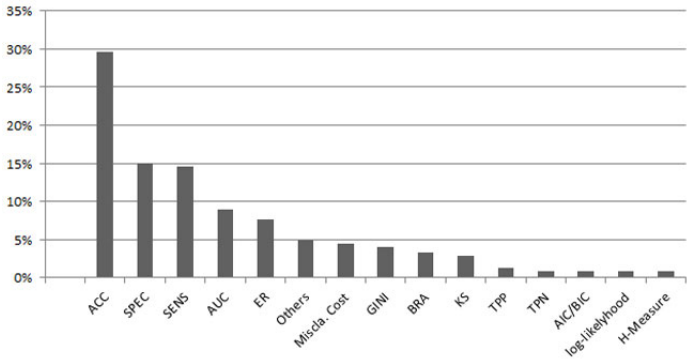


Figura 4.3: Medidas de performance

A Figura 4.5 mostra as técnicas mais comuns e a mudança no padrão em cada período. O modelo logístico permanece como um dos mais utilizados, perdendo o posto para a rede neural na imagem (b). Provavelmente, esse modelo é o preferido para problemas de *credit scoring*. Nota-se também o aparecimento de novas técnicas como SVM e rede neural que com o passar do tempo ganharam vários adeptos. Além disso, notamos também que algumas técnicas perderam espaço, como é o caso da análise de discriminante.

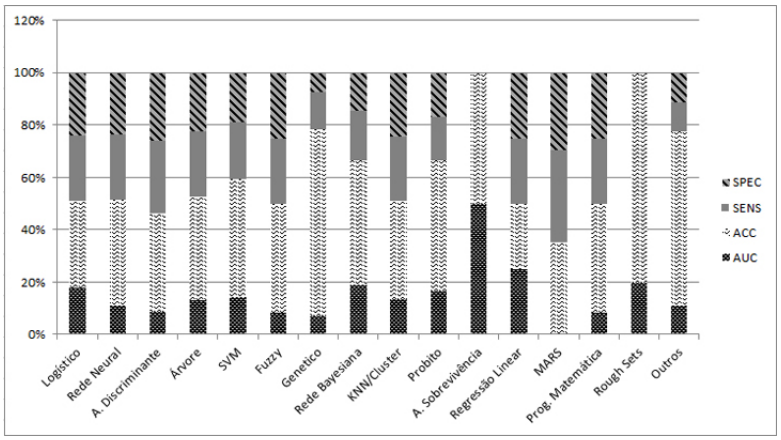
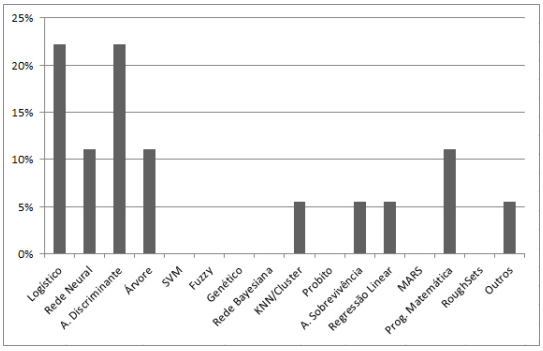
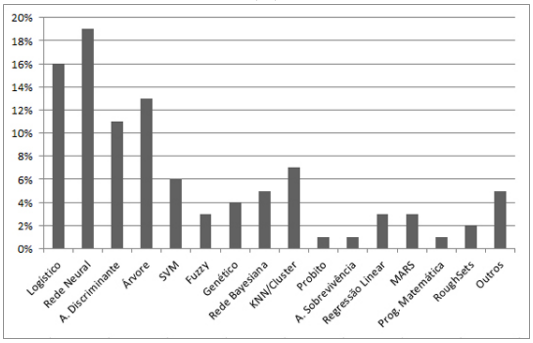


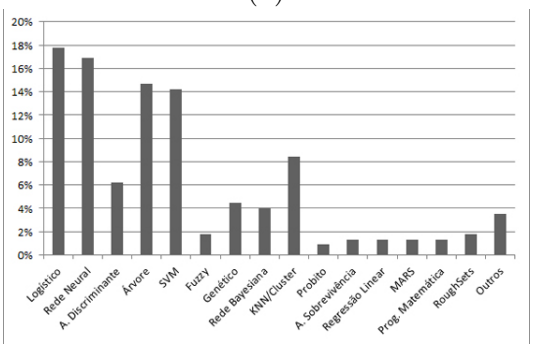
Figura 4.4: Distribuição das medidas em cada técnica



(a)



(b)



(c)

Figura 4.5: (a) [96, 2000]; (b) [2001, 2006]; (c) [2007, 2012]

A Tabela 4.2 exibe uma lista com os autores que mais publicaram no intervalo de tempo considerado neste trabalho. Notamos que Baesens. B aprarece com 4.31%, Vanthienen. J com

3.08% e Gestel. T. V com 2.46%.

Tabela 4.2: Autores mais frequentes.

Autor	%	No
Baesens. B	4.31	14
Vanthienen. J	3.08	10
Gestel. T. V	2.46	8
D. J. Hand	2.46	8
Mues. C	1.85	6
L. C. Thomas	1.85	6
Finlay. S. M	1.23	4
Martens. D	0.92	3
Lai K. K	0.92	3
Lee. T. S	0.92	3
Chiu. C. C	0.92	3
Tsai. C.F	0.92	3
J. N. Crook	0.92	3
Shi. Y	0.92	3
Louzada. F	0.92	3
Wah T. Y.	0.62	2
Ainon R. N.	0.62	2
Lahsasna A.	0.62	2
Others	73.54	239
Total	100	325

O conjunto de técnicas utilizadas em *credit scoring* pode ser dividido em dois segmentos: Técnicas Estatísticas e Outras. Como técnicas estatísticas consideraremos as seguintes metodologias: Regressão logística, análise de discriminante (DA), Probit, Sobrevivência, Regressão Linear e Árvore. E como outras as seguintes técnicas, rede neural, fuzzy, genetic, SVM, roughsets, redes bayesianas, KNN/Cluster, Mars, Mat. Prog., outros.

Vemos na Figura 4.6 que a quantidade de artigos de ambos os segmentos apresentam mudanças bem similares ao longo do tempo, isto é, ambas as curvas apresentam uma tentência de crescimento, cruzam-se várias vezes e, em alguns momentos permanecem próximas. Essa similaridade entre as curvas deve-se ao fato de muitos artigos trabalharem com ambos os segmentos, ou seja, para o mesmo conjunto de dados um modelo estatístico e um outro não estatístico são usados e comparados.

A Figura 4.7 exibe a participação de ambos os segmentos em cada um dos períodos indicados na Figura. Vemos uma ligeira diferença no primeiro período, em que as técnicas estatísticas possuem uma participação maior. Nos dois períodos posteriores a figura exibe uma ligeira vantagem do segundo segmento sobre o primeiro.

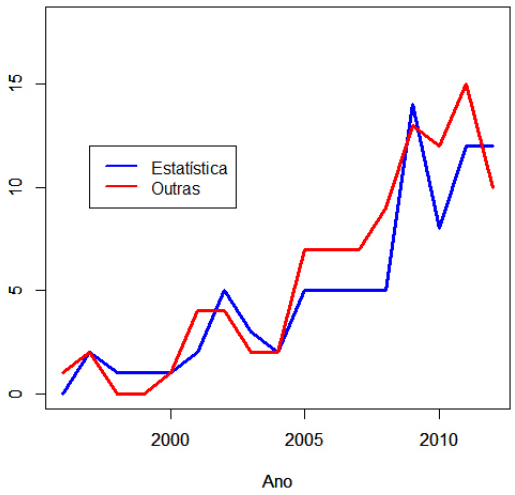


Figura 4.6: Artigos publicados a cada ano

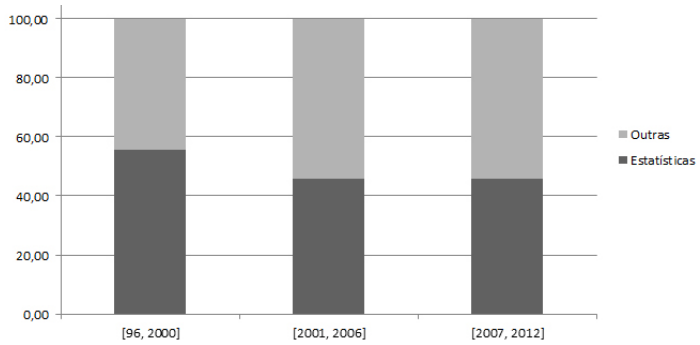


Figura 4.7: Participação de cada grupo em cada período

A Figura 4.8 exibe as medidas de performance mais utilizadas em técnicas estatísticas e com as outras técnicas. Vemos que o padrão é basicamente o mesmo porém, nota-se que nas técnicas não usuais a utilização da medida ACC tem uma participação maior com quase 35% contra algo próximo de 25% nas técnicas estatísticas.

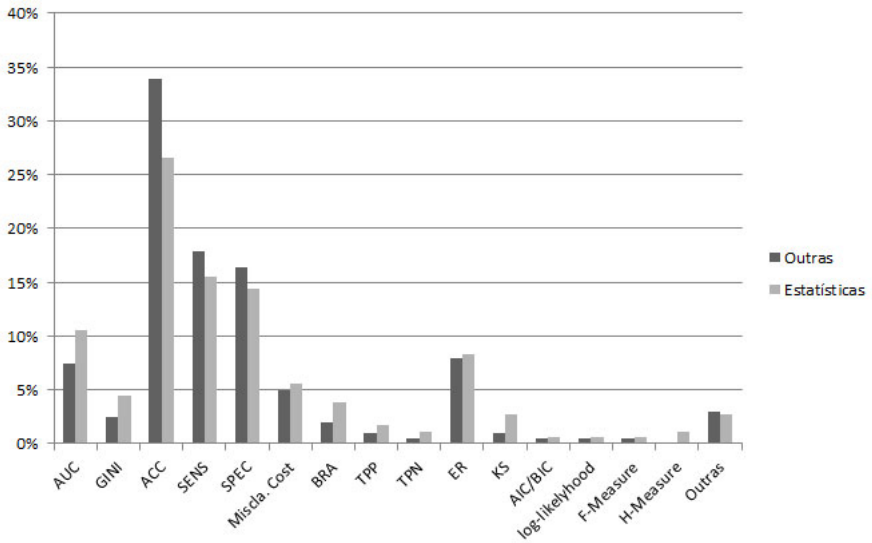


Figura 4.8: Utilização das medidas de performance para cada tipo de técnica.

4.2 Comentários Finais

Este capítulo fez uma análise das técnicas que foram mais empregadas em *credit scoring* no decorrer do intervalo de tempo considerado. Com isso, foi possível obter uma visão geral do cenário e, concluímos que muitas das técnicas empregadas são não estatísticas tais como, redes neurais, SVM e programação matemática. Isso mostra a característica multidisciplinar do problema, que pode ser abordado de um ponto de vista estatístico que, geralmente, utiliza distribuições de probabilidade para a variável resposta, ou também uma abordagem em que esse tipo de suposição não é necessária.

Capítulo 5

COMPARAÇÃO ENTRE AS TRÊS TÉCNICAS MAIS UTILIZADAS

No capítulo 4, vimos que existem várias técnicas utilizadas em problemas de *credit scoring*, sendo as três mais utilizadas, em todo o período, as técnicas de regressão logística, redes neurais e árvores de decisão. Então, responderemos a seguinte pergunta: Dentre essas três técnicas qual foi a que apresentou a melhor performance?

É sabido que não existe uma técnica que suplanta todas as outras, a performance de cada técnica depende da base de dados utilizada, isto é, dadas duas técnicas A e B para uma determinada base de dados, A pode obter uma performance melhor que B e, para uma outra base de dados, B poderá obter uma performance superior que A. O objetivo aqui não é dar a palavra final sobre a melhor técnica.

O objetivo é observar nos artigos utilizados na revisão bibliografica sistemática aqueles que utilizaram regressão logística, redes neurais e árvores de decisão qual foi a performance obtida para essas técnicas e então, fornecer subsídios a uma ideia intuitiva de qual técnica seria a mais promissora no contexto de *credit scoring*.

Em muitas situações não é apenas a performance que importa, sendo que a palavra performance aqui deve ser entendida como capacidade preditiva medida através das várias maneiras possíveis que foram descritas no capítulo 3. Além disso, em certas ocasiões a interpretabilidade do modelo é importante e pode acontecer de um modelo ser mais interpretável que outro, porém aqui analisaremos somente as medidas de capacidade preditiva obtidas.

5.1 Estudo de comparação

Para fazer esse estudo foram selecionados os artigos que possuiam entre as técnicas utilizadas as três mais comuns e então, foram observados os valores obtidos para as três medidas preditivas

mais comuns. Foram selecionados 90 artigos com esse critério.

Na Tabela 5.1 são exibidas as médias das medidas preditivas obtidas com os modelos logístico, rede neural e árvore. A tabela exibe a acurácia(ACC), área sob a curva ROC (AUC), $P(b|B)$ que é a probabilidade estimada do modelo classificar como bom (b) o cliente que é de fato bom (B) e $P(m|M)$ que é a probabilidade estimada do modelo classificar como mau (m) o cliente que é de fato mau (M).

As medidas $P(b|B)$ e $P(m|M)$ nada mais são do que a sensibilidade e a especificidade, porém para definir especificidade e sensibilidade é preciso definir o evento que se está modelando, por exemplo, se o evento modelado for inadimplência a sensibilidade seria $P(m|M)$ e a especificidade seria $P(b|B)$. Entretanto, o evento modelado varia de um trabalho para outro, portanto para padronizar não será definido um evento aqui em vez disso chamaremos de $P(b|B)$ e $P(m|M)$.

	Reg. Log.	Rede Neural	Árvore
ACC	0,783	0,794	0,780
$P(b\|B)$	0,810	0,802	0,798
$P(m\|M)$	0,597	0,662	0,700
AUC	0,781	0,783	0,719

Tabela 5.1: Média das medidas de performance.

A Tabela 5.1 exibe os modelos levando em conta ACC percebemos que nenhum modelo leva grande vantagem sobre os outros. Com relação a essa medida a rede neural possui ligeira vantagem sobre o modelo logístico e árvore.

Por outro lado, se compararmos a probabilidade estimada de acertos entre os bons clientes ($P(b|B)$) o modelo logístico é o que possui o maior valor seguido pela rede neural e árvore. A rede neural volta a possuir o maior valor quando comparado através da área sob a curva ROC (AUC) seguida por logístico e árvore. Um argumento comumente utilizado na literatura que encoraja o uso do AUC é que esse valor é obtido sem considerar um único ponto de corte e por isso poderia avaliar o modelo de forma mais ampla, Gestel. T.V, Baesens. B, Suykens. J. A. K, Poel. D. V, Baestaens. D.E, Willekens. M, 2005.

A árvore de decisão obteve melhor performance quando o quesito foi a probabilidade de acertos entre os maus clientes seguida pela rede neural e logístico.

Nas tabelas 5.2, 5.3 e 5.4 são exibidos primeiro quartil, mediana, média e terceiro quartil das medidas de performance para cada uma das técnicas regressão logística, rede neural e árvore de decisão, respectivamente.

Através das tabelas é possível ter alguma ideia da variação dessas medidas como, por exemplo, se compararmos a ACC do modelo logístico com a ACC da rede neural vemos que o primeiro quartil da rede neural está mais a direita do que o primeiro quartil da regressão logística. Porém, o terceiro quartil da regressão logística está mais a direita do que o mesmo valor para a rede

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,740	0,791	0,783	0,861
$P(b\|B)$	0,763	0,867	0,810	0,895
$P(m\|M)$	0,486	0,645	0,597	0,712
AUC	0,734	0,816	0,781	0,842

Tabela 5.2: Regressão logística

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,742	0,807	0,794	0,835
$P(b\|B)$	0,743	0,845	0,802	0,888
$P(m\|M)$	0,559	0,721	0,662	0,783
AUC	0,726	0,806	0,783	0,852

Tabela 5.3: Rede neural

neural, de maneira análoga é possível proceder para as outras medidas.

Para auxiliar na comparação serão apresentados a seguir os *box plots* para cada uma das técnicas.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,719	0,782	0,780	0,818
$P(b\|B)$	0,793	0,825	0,798	0,908
$P(m\|M)$	0,646	0,709	0,700	0,819
AUC	0,662	0,739	0,719	0,784

Tabela 5.4: Árvore

Nas Figuras 5.1, 5.2, 5.3 e 5.4, são apresentados os *box plots* comparando as medidas ACC, $P(b|B)$, $P(m|M)$ e AUC para cada uma das três técnicas comparadas neste capítulo. O *box plot* é uma ferramenta muito útil para ajudar na visualização dos dados, essa ferramenta fornece uma ideia da posição, dispersão, assimetria, caudas e dados discrepantes, esse tipo de visualização pode ajudar a entender o comportamento dos dados.

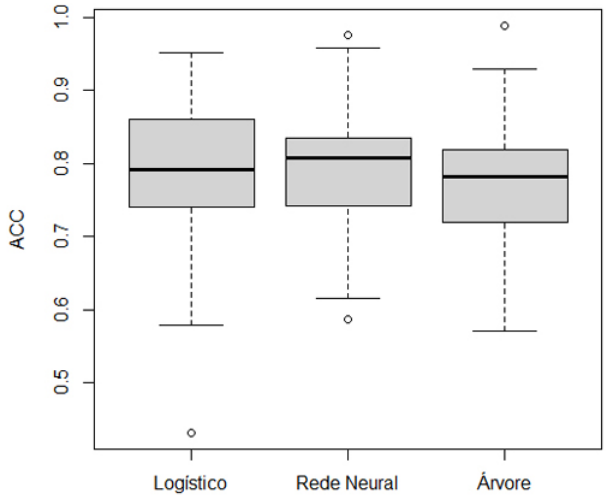


Figura 5.1: *Box plot* comparando ACC

Na Figura 5.1 vemos que o *box plot* da rede neural e do modelo logístico coincidem na parte inferior porém, a mediana para a rede neural está acima de todas as outras medianas. Além disso, a figura indica que a rede neural possui valores menos dispersos para ACC e, o modelo de árvores aparece em desvantagem porque seu *box plot* está delocado para baixo em comparação com as outras duas técnicas.

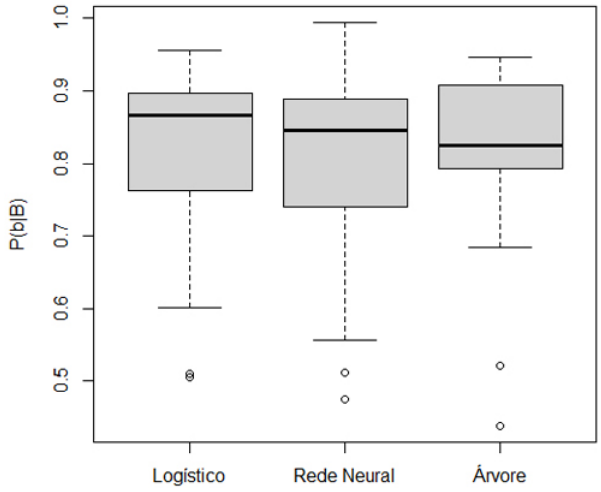


Figura 5.2: *Box plot* comparando $P(b|B)$

Na figura 5.2 observamos que a caixa do *box plot* para a rede neural é maior que as demais, é um indicativo de uma maior dispersão dos valores obtidos por esse modelo. O modelo logístico possui a maior mediana e a árvore de decisão possui uma caixa mais compacta sinalizando para uma dispersão menor e, a caixa do *box plot* para o modelo árvore está ligeiramente mais elevada que a do modelo logístico, porém a mediana do logístico é maior que para a árvore.

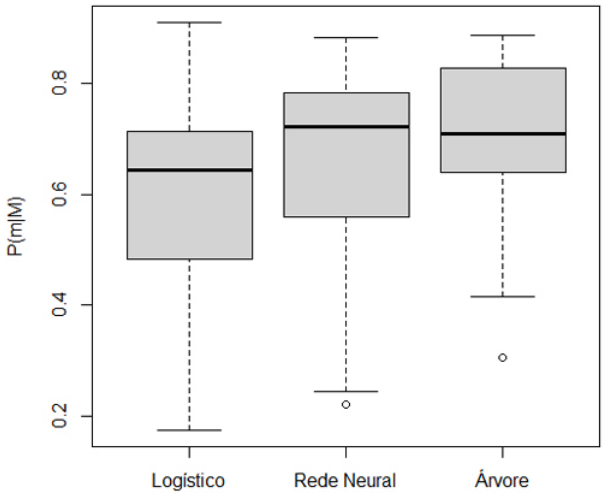


Figura 5.3: *Box plot* comparando $P(m|M)$

A Figura 5.3 exibe os box plot para os valores da probabilidade estimada $P(m|M)$. A figura mostra que a árvore de decisão possui um box plot posicionado na parte mais alta sugerindo uma tendência de valores mais elevados para essa medida. Notamos também pela imagem que a árvore possui uma caixa menor o que sugere uma dispersão menor em comparação com os outros modelos. Nota-se que ora uma determinada técnica leva vantagem para uma determinada medida, ora uma outra técnica apresenta resultados melhores. Portanto, a tarefa de decidir qual a melhor técnica pode ser muito complicada sendo mais prudente testar vários modelos antes de tomar uma decisão.

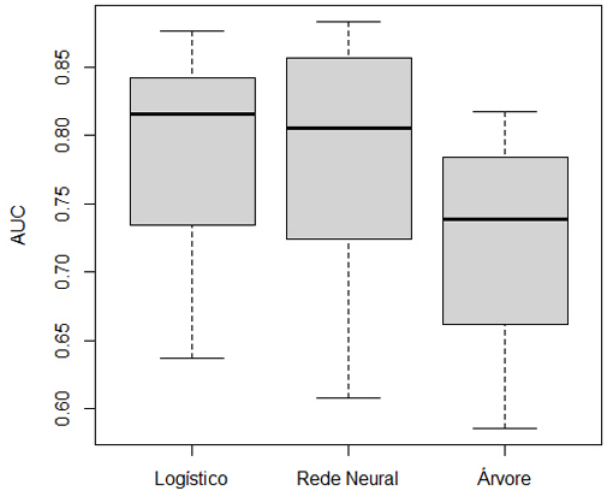


Figura 5.4: *Box plot* comparando AUC

Na Figura 5.4 são exibidos os *box plots* com os resultados das medidas da área sob a curva *ROC* (AUC). Nesse quesito o modelo de árvore aparenta ter uma performance mais pobre em comparação com os outros modelos. As redes neurais e o modelo logístico apresentaram resultados parecidos diferindo nos seguintes pontos: a mediana para o modelo logístico está situada acima da mediana obtida pela rede neural e a caixa do *box plot* para o modelo logístico tem um tamanho menor do que o obtido pela rede neural indicativo de uma menor dispersão.

Os resultados mostram que o melhor a ser feito no momento de escolher qual técnica utilizar é testar várias e escolher a que melhor se adequar ao problema. As redes neurais são muito boas para obter alta capacidade de acertos, porém deixa a desejar no quesito interpretação e o tempo de treino pode ser longo. As árvores de decisão também apresentaram resultados interessantes com uma certa desvantagem no AUC, as árvores dividem o conjunto de variáveis explicativas em conjuntos disjuntos e por conta disso podem ser utilizadas para criar *clusters*. Para a regressão logística é muito comum criar faixas de escore e, uma vez que se define a faixa de escore de um cliente, define-se as condições da transação do cliente junto a instituição financeira.

5.2 Comentários Finais

Existem inumeras técnicas e modelos que podem ser aplicados em problemas de classificação. A escolha da técnica envolve várias etapas como, por exemplo, conhecer a variável resposta, isto é, continua, contínua ou discreta e, se no caso contínuo, em qual(is) intervalo(s) está definida. E, no caso discreto, se é binário ou não.

Existem técnicas que são desenvolvidas pensando em casos específicos como, por exemplo, resposta binária, e existem técnicas elaboradas para tratar casos mais geral de resposta.

Os resultados mostram que, o melhor a ser feito é testar as várias técnicas no conjunto de dados e então, escolher a que forneceu o desempenho mais satisfatório.

Capítulo 6

ESTUDO DE SIMULAÇÃO

Nesta seção faremos uma comparação entre as técnicas através de simulação, englobando as três técnicas mais utilizadas e algumas outras para que o estudo fique mais amplo. Na próxima parte deste capítulo, será formalizado como os dados foram simulados e as ferramentas utilizadas para realizar esse estudo.

6.1 Simulação

O estudo foi realizado em uma base de dados simulada em que a proporção de maus (evento) e bons (não evento) varia da seguinte forma 10%-90%, 20%-80%, 30%-70%, 40%-60% e 50%-50%, respectivamente. As medidas calculadas foram ACC, Sensibilidade = $P(m|M)$, Especificidade = $P(b|B)$ e AUC. Várias técnicas foram testadas: árvore, rede neural, regressão logística, análise de discriminante linear (LDA), análise de discriminante quadrática (QDA), *naive* Bayes e *Support Vector Machine* (SVM).

O software utilizado para fazer os testes foi o R e os pacotes utilizados foram, rpart, nnet, MASS, e1071 e AUC que contém as funções de ajuste de árvore, rede neural, LDA, QDA, logítica e as rotinas para o cáculo da medida AUC. Ao fazer o juste das redes neurais e árvores é possível especificar para que a saída seja categórica, no caso da regressão logística a saída é um valor no intervalo $(0, 1)$ e por isso é preciso definir pontos de corte e por isso as medidas da regressão logística são obtidas variando os cortes em 0.3, 0.4 e 0.5. É importante dizer que é comum encontrar o termo treinamento para se referir, por exemplo, ao ajuste de redes neurais e árvores e o termo ajuste é mais comum quando se trata de modelos de regressão. Entretanto, não faremos distinção entre ajuste e treinamento.

Para obter os dados, consideramos o procedimento descrito em Breiman (1998): as covariáveis dos bons clientes seguem uma normal multivariada com dimensão 20, com vetor de médias zero e matriz de variâncias $4 * I_{20}$ que é a matriz identidade com sua diagonal multipli-

cada por 4. As covariáveis dos maus clientes seguem uma distribuição normal multivariada com dimensão 20 com vetor de médias 1/sqrt(20) e matriz de variância igual à identidade. Para realizar o ajuste foram simulados 100.000 (cem mil) observações e para calcular as medidas foram simuladas mais 20.000 (vinte mil) observações, ou seja, os modelos são testados em dados que não foram utilizados para o ajuste. Além disso, na base de teste os dados estão balanceados, isto é, metade são eventos e metade não evento além disso, todas as covariáveis são contínuas e não foram categorizadas. Para deixar mais claro como foi realizada a simulação, descrevemos um passo-a-passo.

Primeiro passo é feita a simulação da base utilizada no ajuste com 100.000 (cem mil) observações de modo que 10% são eventos e 90% não eventos, isto é, 10% são considerados maus pagadores e 90% são bons pagadores. Então, ajusta-se o modelo.

Segundo passo é simular a base de teste com 20.000 (vinte mil) observações de modo que metade é evento e a outra metade não evento.

Terceiro passo é utilizar a base de teste para calcular as medidas, e repetir os passos acima 100 vezes.

Concluídas as 100 repetições, a proporção de eventos e não eventos é alterada e o procedimento é repetido.

As Tabelas apresentadas a seguir, 6.1, 6.2, 6.3, 6.4 e 6.5 apresentam o resumo das medidas obtidas para árvore de decisão para cada uma das bases com proporcões de eventos e não eventos variadas.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,7358	0,7406	0,7408	0,7458
$P(b\|B)$	0,9872	0,9886	0,9883	0,9893
$P(m\|M)$	0,4824	0,4934	0,4932	0,5043

Tabela 6.1: Árvore base 10-90

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,7838	0,7880	0,7879	0,7923
$P(b\|B)$	0,9711	0,9730	0,9726	0,9744
$P(m\|M)$	0,5952	0,6044	0,6032	0,6132

Tabela 6.2: Árvore base 20-80

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,8117	0,8149	0,8152	0,8186
$P(b\|B)$	0,9502	0,9524	0,9522	0,9544
$P(m\|M)$	0,6702	0,6767	0,6782	0,6852

Tabela 6.3: Árvore base 30-70

No experimento realizado percebemos que o algoritmo teve uma performance melhor a medida que a base se torna mais equilibrada. Observamos nas Tabelas 6.5 e 6.1 em que o desempenho sofre uma mudança. Na Tabela 6.1 o algoritmo possui uma performance boa para detectar bons clientes porém, a ACC não é tão elevada e a capacidade do algoritmo de detectar eventos dado que o evento ocorreu (sensibilidade) também não é muito satisfatória.

A medida que a base utilizada no ajuste vai se tornando mais balanceada o algoritmo apresenta uma performance melhor, a Tabela 6.5 exibe essa melhora. Os valores para $P(m|M)$ apresentam um crescimento considerável e os valores de ACC também aumentaram. Entretanto, houve uma leve queda nos valores de $P(b|B)$. É possível notar um desempenho mais parcimonioso já na Tabela 6.4 em que a base de dados já apresenta uma quantidade grande de eventos e não eventos permitindo que o algoritmo pudesse capturar as características de cada grupo. No caso da árvore, não foi preciso definir um ponto de corte para calcular as medidas ACC, $P(b|B)$ e $P(m|M)$, pois o *software* fornecia as saídas em categorias assim, foi utilizada a saída contínua somente para calcular o AUC que será apresentado mais adiante.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,8294	0,8326	0,8329	0,8370
$P(b\|B)$	0,9224	0,9260	0,9255	0,9289
$P(m\|M)$	0,7331	0,74	0,7403	0,748

Tabela 6.4: Árvore base 40-60

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,8381	0,8408	0,8411	0,8434
$P(b\|B)$	0,8819	0,8856	0,8855	0,8892
$P(m\|M)$	0,7918	0,7961	0,7967	0,8013

Tabela 6.5: Árvore base 50-50

A Tabela 6.6 exibe as medidas de AUC obtidas variando as proporções de evento e não evento.

AUC	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
10%-90%	0,8272	0,8316	0,8311	0,8350
20%-80%	0,8394	0,8432	0,8435	0,8477
30%-70%	0,8498	0,8534	0,8533	0,8564
40%-60%	0,8575	0,8612	0,8612	0,8655
50%-50%	0,8666	0,8698	0,8697	0,8727

Tabela 6.6: Medidas AUC - Árvore

A Tabela 6.6 apresenta o resumo das medidas obtidas para a área sob a curva ROC (AUC). Notamos que os valores permanecem, na casa dos 80% para todas as bases utilizadas e, da mesma forma que os resultados anteriores os resultados apresentam uma melhora para quando os dados ficam mais equilibrados. Além disso, observamos que não houve grandes variações, isto é, os valores das tabelas exibidas acima não apresentam grandes variações dentro de cada tabela.

A seguir, as Tabelas exibem os resultados obtidos para o modelo logístico. Os pontos de corte adotados foram 0,3; 0,4; 0,5. Para cada ponto de corte foram feitas simulações variando as proporções de bons e maus na base. Aqui é necessário definir um corte pois o modelo logístico fornece a probabilidade de ocorrer o evento, e os eventos foram definidos da seguinte forma, mau pagador $y = 1$ caso contrário $y = 0$.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,498	0,4983	0,4983	0,4985
$P(b\|B)$	0,9961	0,9966	0,9965	0,997
$P(m\|M)$	0	0	$4 * 10^{-6}$	0

Tabela 6.7: Logístico base 10-90, corte 0,3

A Tabela 6.7 apresenta o resumo das medidas para uma base com 10% de eventos e um ponto de corte 0,3. Ou seja, se a probabilidade fornecida pelo modelo for maior que 0,3 então o indivíduo é considerado evento. O resultado não é satisfatório, esse modelo não foi capaz de detectar se um cliente é mau pagador, sabendo-se que, na realidade, ele é um mau pagador, fato que impacta em uma ACC baixa. Embora o valor de $P(b|B)$ seja alto, esse modelo certamente não é adequado.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,4998	0,4999	0,4999	0,5
$P(b\|B)$	0,9997	0,9998	0,9997	0,9999
$P(m\|M)$	0	0	0	0

Tabela 6.8: Logístico base 10-90, corte 0,4

Em 6.8 e 6.9 o ponto de corte é mudado para 0,4 e 0,5, respectivamente. Os resultados são similares aos fornecidos na Tabela 6.7. As conclusões são as mesmas, o modelo não é satisfatório. Para esse conjunto de dados o modelo logístico parece não ser muito interessante. Cabe aqui dizer que, testamos apenas para 3 valores de pontos de corte e que talvez seria possível obter valores melhores para esse desbalanceamento da base adotando um ponto de corte mais baixo aos que foram aqui apresentados.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,5	0,5	0,5	0,5
$P(b\|B)$	1	1	1	1
$P(m\|M)$	0	0	0	0

Tabela 6.9: Logístico base 10-90, corte 0,5

Vamos agora fazer o mesmo procedimento com uma base de dados que possua uma proporção maior de eventos.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,4854	0,4868	0,4869	0,4881
$P(b\|B)$	0,8702	0,8730	0,8726	0,8751
$P(m\|M)$	0,0977	0,1010	0,1011	0,1046

Tabela 6.10: Logístico base 20-80, corte 0,3

A Tabela 6.10 apresenta valores melhores do que as anteriores, os valores de $P(m|M)$ são maiores porém houve uma queda no valores de $P(b|B)$ e, os valores de ACC continuam basicamente os mesmos em torno de 0,5. É ainda um modelo insatisfatório pois possui valores de ACC e $P(m|M)$ muito baixos.

Os valores maiores de $P(m|M)$ indicam o que já era intuitivamente esperado, com uma base de dados com uma quantidade maior de eventos o modelo foi capaz de capturar melhor as características deste grupo.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,4850	0,4858	0,4859	0,4865
$P(b\|B)$	0,9670	0,9686	0,9685	0,9698
$P(m\|M)$	0.002875	0.0032	0.003292	0.0037

Tabela 6.11: Logístico base 20-80, corte 0,4

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,4968	0,4971	0,4970	0,4973
$P(b\|B)$	0,9935	0,9942	0,9941	0,9946
$P(m\|M)$	0	0	$1,6 * 10^{-5}$	0

Tabela 6.12: Logístico base 20-80, corte 0,5

As Tabelas 6.11 e 6.12 apresentam os resultados para os cortes 0,4 e 0,5. Notamos que a medida que o ponto de corte aumenta o modelo tem uma capacidade pior para e medida $P(m|M)$.Vamos aumentar a quantidade de eventos na base de ajuste e avaliar o comportamento do modelo.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,6418	0,6438	0,6439	0,6458
$P(b\|B)$	0,5983	0,6006	0,6012	0,6039
$P(m\|M)$	0,6824	0,6862	0,6866	0,6894

Tabela 6.13: Logístico base 30-70, corte 0,3

Em 6.13 observamos, que o modelo logístico apresentou um desempenho bem melhor do que anteriormente. Embora o valor de $P(b|B)$ tenha caído bastante, a capacidade de detectar maus clientes (evento) dado que o evento ocorreu $P(m|M)$ é bem superior neste caso. Para esse cenário, o valor de ACC também apresentou aumento antes a taxa de acerto do modelo era em torno de 50%. Porém, esse desempenho ainda é inferior ao apresentado pela árvore mostrado anteriormente.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,5109	0,5123	0,5125	0,5144
$P(b\|B)$	0,8162	0,8192	0,8193	0,8220
$P(m\|M)$	0,2009	0,2048	0,2056	0,2099

Tabela 6.14: Logístico base 30-70, corte 0,4

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,4771	0,4779	0,478	0,4792
$P(b\|B)$	0,9326	0,9346	0,9346	0,9365
$P(m\|M)$	0,0204	0,02145	0,02147	0,0227

Tabela 6.15: Logístico base 30-70, corte 0,5

Em 6.14 e 6.15 variamos o ponto de corte e é fácil perceber que o desempenho em relação as medidas $P(m|M)$ e ACC sofre uma deterioração. Porém, com relação a medida $P(b|B)$ obtemos uma melhora. O custo dessa melhora é muito alto uma vez que, em geral, o custo de aceitar um cliente mau (*default*) é maior do que recusar um cliente bom.

Vamos passar para o próximo caso em que a base de ajuste é quase totalmente equilibrada.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,6589	0,6608	0,6609	0,6623
$P(b\|B)$	0,3657	0,3698	0,3697	0,3733
$P(m\|M)$	0,9501	0,9523	0,9521	0,9539

Tabela 6.16: Logístico base 40-60, corte 0,3

É bastante interessante visualizar o comportamento do modelo em vários cenários possíveis. Na Tabela 6.16 observamos uma certa inversão de comportamento, até o momento as medidas $P(b|B)$ eram maiores do que as $P(m|M)$ agora vemos que isso se inverteu. O modelo agora está detectando eventos com maior facilidade porém, como é de praxe, quando uma melhora aparece em uma das medidas a outra tende a cair.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,6409	0,643	0,643	0,6454
$P(b\|B)$	0,5994	0,6015	0,602	0,605
$P(m\|M)$	0,6811	0,6846	0,684	0,6871

Tabela 6.17: Logístico base 40-60, corte 0,4

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,5284	0,5308	0,5306	0,5322
$P(b\|B)$	0,7847	0,789	0,7888	0,7921
$P(m\|M)$	0,2681	0,2724	0,2724	0,276

Tabela 6.18: Logístico base 40-60, corte 0,5

Nota-se também que a medida que os dados tornam-se mais balanceados os resultados tornam-se mais aceitáveis como podemos ver nas tabelas 6.17 e 6.18. Embora o comportamento do modelo logístico venha apresentando uma melhora, seu desempenho ficou abaixo da árvore. No caso do logístico o valor de ACC, que é a taxa geral de acerto, nunca chegou a 70%.

Iremos agora ver o comportamento do modelo para o caso de uma base de ajuste totalmente balanceada. É possível argumentar que em um banco de dados de *credit scoring* uma proporção de 10% de *default* é exageradamente elevada tornando assim esse estudo de simulação muito distante da realidade. Porém, o objetivo aqui é analisar o comportamento das três técnicas mais populares, de acordo com o levantamento feito, e compará-las segundo as medidas de performance mais utilizadas.

Nas Tabelas 6.19, 6.20 e 6.21 são exibidos os resultados para uma base balanceada. Na Tabela 6.19 o modelo apresenta um desempenho razoável para a medida ACC, em relação ao foi apresentado anteriormente. O valor de $P(m|M)$ está bastante elevado e por conta disso houve uma queda muito grande nos valores de $P(b|B)$ o que certamente inviabilizaria o modelo pois está rejeitando muitos indivíduos não *default*.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,6065	0,608	0,6081	0,6095
$P(b\|B)$	0,2192	0,2216	0,2216	0,2245
$P(m\|M)$	0,9940	0,9945	0,9945	0,995

Tabela 6.19: Logístico base 50-50, corte 0,3

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,6664	0,6689	0,6685	0,6705
$P(b\|B)$	0,4053	0,4093	0,409	0,4131
$P(m\|M)$	0,9264	0,9282	0,9281	0,9302

Tabela 6.20: Logístico base 50-50, corte 0,4

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,64	0,6428	0,6426	0,6449
$P(b\|B)$	0,6	0,6034	0,6032	0,607
$P(m\|M)$	0,6793	0,6821	0,6821	0,685

Tabela 6.21: Logístico base 50-50, corte 0,5

A Tabela 6.21 apresenta o resultado mais parcimonioso para esse modelo e, também, os valores estão muito próximos do melhores valores obtidos até aqui entretanto, essa opinião pode ser diferente para um outro indivíduo. Embora os resultados em 6.21 sejam "satisfatórios", isto é, valores equilibrados para $P(b|B)$ e $P(m|M)$, a taxa total de acerto, ACC, está baixa com mais de 30% de erro.

AUC	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
10%-90%	0,6699	0,6722	0,6720	0,6750
20%-80%	0,6707	0,6721	0,6727	0,6751
30%-70%	0,67	0,6717	0,6724	0,6746
40%-60%	0,6704	0,6729	0,6731	0,676
50%-50%	0,6701	0,673	0,6728	0,6758

Tabela 6.22: Medidas AUC - Logístico

A Tabela 6.22 apresenta as medidas de AUC, nota-se que não há muita diferença entre as medidas com a variação da proporção de eventos e não eventos na base. Isso reforça a ideia de que nas tabelas acima em que o modelo logístico teve um desempenho muito ruim poderia ser melhorado com uma investigação mais minuciosa do ponto de corte mais adequado para cada caso. Entretanto, notamos também que o modelo de árvore teve um desempenho melhor para esse tipo de base de dados, como é possível constatar na tabela 6.6.

Os resultados abaixo foram obtidos através das simulações de redes neurais. Para realizar as simulações utilizamos o pacote *nnet* do *software* R. Constatamos também que uma desvatagem relacionada a rede neural é seu tempo de treinamento que pode ser muito longo, essa característica é mencionada na literatura.

O treinamento da rede neural foi realizado sob três configurações distintas, isto é, primeiro foi ajustado um modelo com três elementos na camada interna (*hidden layer*), posteriormante com seis na camada interna e com 9 elementos na camada interna. As Tabelas 6.23, 6.24 e 6.25 exibem os resultados obtidos com uma base de treinamento com 10% de *default* e 90% de não *default*.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,5	0,5	0,5	0,5
$P(b\|B)$	1	1	1	1
$P(m\|M)$	0	0	0	0

Tabela 6.23: Rede neural 10-90, camada interna 3.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,6958	0,7017	0,7029	0,7177
$P(b\|B)$	0,9666	0,9681	0,9683	0,9694
$P(m\|M)$	0,4232	0,4358	0,4375	0,469

Tabela 6.24: Rede neural 10-90, camada interna 6.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,7856	0,7897	0,7926	0,8031
$P(b\|B)$	0,8031	0,9706	0,9704	0,9716
$P(m\|M)$	0,6005	0,6108	0,6148	0,6362

Tabela 6.25: Rede neural 10-90, camada interna 9.

Com três elementos na camada interna notamos que o algoritmo não obteve um bom desempenho, o algoritmo obteve um valor muito baixo para ACC e não foi capaz de detectar os clientes caracterizados como *default*. Com seis elementos na camada interna o resultado é mais satisfatório, notamos um aumento no valor de ACC e, além disso, o algoritmo foi capaz de detectar uma parte dos *default*, notamos que o algoritmo consegue absorver melhor as características dos não *default*. Isso, provavelmente, deve-se ao fato da base de treinamento estar desbalanceada com muito mais observações que não são *default*. Na Tabela 6.25 observamos que a tendência de melhora na performance continua.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,6139	0,6163	0,6078	0,6194
$P(b\|B)$	0,9233	0,9253	0,9309	0,9267
$P(m\|M)$	0,3024	0,3092	0,2847	0,3154

Tabela 6.26: Rede neural 20-80, camada interna 3.

Na tabela 6.26 os resultados são mais equilibrados, notamos que as redes com três elementos na camada interna foi capaz de detectar alguns clientes caracterizados como *default*. Porém, como já era esperado, a classe que possui mais observações na base de dados de treinamento apresenta maior probabilidade de ser detectada, isto é, a Tabela mostra que é mais fácil detectar não *default* do que os *default*.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,7919	0,805	0,801	0,8099
$P(b\|B)$	0,9255	0,9278	0,9277	0,93
$P(m\|M)$	0,6577	0,684	0,6743	0,6913

Tabela 6.27: Rede neural 20-80, camada interna 6.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,8535	0,8629	0,8629	0,8666
$P(b\|B)$	0,9422	0,9451	0,9450	0,9473
$P(m\|M)$	0,7642	0,7802	0,7775	0,7775

Tabela 6.28: Rede neural 20-80, camada interna 9.

Em 6.27, com seis camadas o resultado apresenta uma melhora considerável. Com relação os resultados apresentados em 6.26, houve evolução em todas as medidas e, na Tabela 6.28, observamos um desempenho ainda melhor. É possível notar também uma evolução no desempenho do algoritmo a medida em que as bases tonam-se mais balanceadas e, com o aumento no número de elementos na camada interna.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,7252	0,7278	0,7201	0,7301
$P(b\|B)$	0,8313	0,8342	0,8326	0,8369
$P(m\|M)$	0,6166	0,6206	0,6077	0,6251

Tabela 6.29: Rede neural 30-70, camada interna 3.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,8337	0,8442	0,8408	0,8469
$P(b\|B)$	0,8906	0,8934	0,8926	0,896
$P(m\|M)$	0,7765	0,7947	0,7891	0,7996

Tabela 6.30: Rede neural 30-70, camada interna 6.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,8819	0,8868	0,8861	0,8912
$P(b\|B)$	0,9187	0,9218	0,9215	0,9252
$P(m\|M)$	0,8448	0,8448	0,8508	0,8586

Tabela 6.31: Rede neural 30-70, camada interna 9.

Em 6.29 e 6.30 apresentam os resultados obtidos para uma base de treinamento com 30% de eventos e 70% não eventos. Como tendência geral, é possível dizer que a rede neural com mais elementos na camada interna, em geral, obterá melhor performance. Porém, a medida que os elementos da camada interna aumenta, o tempo de treinamento também aumenta. A seguir apresentamos mais alguns resultados obtidos por simulação.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,7618	0,7645	0,7619	0,7665
$P(b\|B)$	0,7565	0,7604	0,7547	0,7645
$P(m\|M)$	0,7649	0,7688	0,7691	0,773

Tabela 6.32: Rede neural 40-60, camada interna 3.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,8571	0,8603	0,8582	0,8626
$P(b\|B)$	0,8562	0,8588	0,8579	0,8617
$P(m\|M)$	0,8564	0,8615	0,8584	0,8642

Tabela 6.33: Rede neural 40-60, camada interna 6.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,8943	0,8968	0,8958	0,8993
$P(b\|B)$	0,8945	0,8980	0,8976	0,9014
$P(m\|M)$	0,8911	0,8948	0,8940	0,8993

Tabela 6.34: Rede neural 40-60, camada interna 9.

Notamos que em 6.32 os resultados são mais aceitáveis do que aqueles apresentados com um desbalanceamento de 10% e 90%, ou seja, a medida que a base se torna mais balanceada a rede com 3 elementos obtém uma performance melhor. No caso em que temos proporções como apresentadas no início (10% e 90%) poderíamos ser levados a pensar que a rede com 3 elementos não poderia ser utilizada. Porém, em caso de uma base mais parcimoniosa, é possível utilizar tal configuração. E, como já era intuitivamente esperado, em 6.33 e 6.34 com seis e nove elementos na camada interna a rede neural apresenta melhor performance.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,7724	0,7743	0,7724	0,7761
$P(b\|B)$	0,6944	0,6971	0,6928	0,6995
$P(m\|M)$	0,849	0,8518	0,852	0,854

Tabela 6.35: Rede neural 50-50, camada interna 3.

Em 6.35 e 6.36 os dados estão equilibrados na base de treino e nesse caso em 6.35 o algoritmo teve mais facilidade em detectar *default*, embora as quantidades de observações de cada classe são iguais. Na Tabela 6.36 os resultados mostram que a rede neural possui uma capacidade mais equilibrada para separar cada classe, além disso, a performance em 6.36 e 6.37 é superior. Como já era esperado.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,8549	0,8645	0,861	0,8673
$P(b\|B)$	0,8166	0,8259	0,8223	0,8295
$P(m\|M)$	0,8942	0,9022	0,8996	0,9053

Tabela 6.36: Rede neural 50-50, camada interna 6.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,8944	0,9002	0,8988	0,9033
$P(b\|B)$	0,8674	0,8744	0,8733	0,8794
$P(m\|M)$	0,9216	0,9247	0,9243	0,9286

Tabela 6.37: Rede neural 50-50, camada interna 9.

As Tabelas 6.38 e 6.39 exibem os valores de AUC para cada base de dados. Como já foi dito, com mais elementos na camada interna a performance da rede neural tende a melhorar, e o tempo de treinamento também sofre um acréscimo. Essa melhora na performance pode ser constatada nas Tabelas abaixo.

AUC	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
10%-90%	0,8307	0,8327	0,8308	0,8345
20%-80%	0,8311	0,8333	0,8287	0,8354
30%-70%	0,8311	0,8334	0,8277	0,8357
40%-60%	0,8315	0,8334	0,8298	0,8362
50%-50%	0,8321	0,8347	0,8316	0,836

Tabela 6.38: Medidas AUC - Rede neural, camada interna 3.

AUC	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
10%-90%	0,9209	0,9252	0,9255	0,9326
20%-80%	0,923	0,9307	0,9281	0,9331
30%-70%	0,9258	0,932	0,9298	0,9338
40%-60%	0,9298	0,932	0,9306	0,9338
50%-50%	0,9243	0,9317	0,9289	0,9336

Tabela 6.39: Medidas AUC - Rede neural, camada interna 6.

AUC	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
10%-90%	0,9537	0,9558	0,9564	0,9602
20%-80%	0,9559	0,9605	0,9599	0,9630
30%-70%	0,9586	0,9607	0,9603	0,9629
40%-60%	0,9586	0,9604	0,9598	0,9626
50%-50%	0,9559	0,9604	0,9595	0,9621

Tabela 6.40: Medidas AUC - Rede neural, camada interna 9.

Os resultados a seguir foram obtidos utilizando a técnica SVM (*Support Vector Machine*), foi feita uma mudança no conjunto de dados pois o tempo computacional utilizado para treinar esse método é demasiadamente longo. Portanto, utilizamos uma base de treino com (dez mil) observações e não 100.000 (cem mil) como nas situações anteriores e a base de teste que nas técnicas utilizadas até agora possuía 20.000 (vinte mil) foi reduzida para 2.000 (dois mil).

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9675	0,9700	0,9702	0,9735
$P(b\|B)$	0,9910	0,9930	0,9928	0,9950
$P(m\|M)$	0,9430	0,9480	0,9476	0,9530

Tabela 6.41: SVM 10-90.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9775	0,9800	0,9799	0,9821
$P(b\|B)$	0,9910	0,9930	0,9928	0,9950
$P(m\|M)$	0,9430	0,9480	0,9476	0,9530

Tabela 6.42: SVM 20-80.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9800	0,9820	0,9823	0,9845
$P(b\|B)$	0,9830	0,9850	0,9853	0,9880
$P(m\|M)$	0,9760	0,9795	0,9793	0,9830

Tabela 6.43: SVM 30-70.

Como pode ser visto nas Tabelas 6.41, 6.42, 6.43, 6.44 e 6.45 a técnica apresentou um comportamento bastante estável, mantendo o desempenho para as várias situações do banco de dados. Esse comportamento é um ponto bastante positivo para o SVM, como exibido anteriormente como, por exemplo, na técnica rede neural a performance foi melhorando a medida que o banco de dados ficou mais balanceado.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9820	0,9832	0,9835	0,9855
$P(b\|B)$	0,9780	0,9820	0,9813	0,9842
$P(m\|M)$	0,9830	0,9860	0,9857	0,9882

Tabela 6.44: SVM 40-60.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9815	0,9840	0,9837	0,9855
$P(b\|B)$	0,9750	0,9780	0,9779	0,9810
$P(m\|M)$	0,9880	0,9900	0,9895	0,9920

Tabela 6.45: SVM 50-50.

A técnica SVM apresentou resultados estáveis, ou seja, mantendo a performance para os vários casos de desbalanceamento dos dados. Além disso, essa técnica não faz suposições acerca da distribuição de probabilidade dos dados como ocorre, por exemplo, em análise de discriminante linear, análise de discriminante quadrática e naive Bayes. Portanto, é uma boa escolha dentre os vários possíveis métodos de classificação de credit scoring

A desvantagem é que seu tempo de treinamento pode ser mais longo e o modelo possui uma construção teórica mais rebuscada.

AUC	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
10%-90%	0,9980	0,9984	0,9983	0,9987
20%-80%	0,9981	0,9985	0,9985	0,9989
30%-70%	0,9981	0,9985	0,9984	0,9989
40%-60%	0,9978	0,9983	0,9982	0,9986
50%-50%	0,9977	0,9981	0,9981	0,9985

Tabela 6.46: Medidas AUC - Support Vector Machine.

Nos resultados obtidos com LDA, QDA e naive Bayes, voltamos a utilizar os tamanhos originais das bases.

Como será exibido em seguida, a técnica LDA não obteve bons resultados e isso deve-se ao fato da técnica assumir que as matrizes de variâncias são iguais para as classes e sabemos através da simulação que essa suposição vai de encontro as características dos dados. A técnica QDA, por sua vez, não possui essa hipótese e obteve resultados mais promissores.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,5	0,5	0,5	0,5
$P(b\|B)$	1	1	1	1
$P(m\|M)$	0	0	0	0

Tabela 6.47: LDA 10-90.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,4975	0,4978	0,4978	0,4981
$P(b\|B)$	0,9951	0,9956	0,9955	0,9962
$P(m\|M)$	0	0	0	0

Tabela 6.48: LDA 20-80.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,4779	0,4788	0,4789	0,4797
$P(b\|B)$	0,9387	0,9408	0,9406	0,9425
$P(m\|M)$	0,01570	0,01725	0,01710	0,01830

Tabela 6.49: LDA 30-70.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,5260	0,5274	0,5279	0,5300
$P(b\|B)$	0,7894	0,7936	0,7930	0,7964
$P(m\|M)$	0,2588	0,2622	0,2628	0,2670

Tabela 6.50: LDA 40-60.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,6425	0,6445	0,6450	0,6482
$P(b\|B)$	0,5941	0,5972	0,5977	0,6013
$P(m\|M)$	0,01570	0,01725	0,01710	0,01830

Tabela 6.51: LDA 50-50.

AUC	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
10%-90%	0,6705	0,6725	0,6725	0,6744
20%-80%	0,6692	0,6719	0,6722	0,6746
30%-70%	0,6706	0,6728	0,6728	0,6750
40%-60%	0,6700	0,6722	0,6722	0,6747
50%-50%	0,6696	0,6723	0,6723	0,6747

Tabela 6.52: Medidas AUC - LDA.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9705	0,9712	0,9712	0,9719
$P(b\|B)$	0,9931	0,9936	0,9936	0,9941
$P(m\|M)$	0,9476	0,9486	0,9488	0,9499

Tabela 6.53: QDA 10-90.

Como podemos constatar nas Tabelas 6.47, 6.48, 6.49, 6.50, 6.51 e 6.52 o desempenho obtido para LDA não é muito interessante. Embora esse desempenho apresenta uma melhora para os casos de amostras mais balanceadas ainda sim os resultados não são bons e, isso ocorre, como já foi dito, pela suposição de matrizes de variâncias iguais para ambas as classes. A seguir, serão apresentados os resultados obtidos com a técnica QDA que não faz a suposição de matrizes iguais e veremos que os resultados são melhores.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9793	0,9800	0,9800	0,9808
$P(b\|B)$	0,9895	0,9900	0,9900	0,9905
$P(m\|M)$	0,9686	0,9700	0,9701	0,9715

Tabela 6.54: QDA 20-80.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9826	0,9832	0,9832	0,9839
$P(b\|B)$	0,9863	0,9870	0,9871	0,9878
$P(m\|M)$	0,9784	0,9793	0,9794	0,9806

Tabela 6.55: QDA 30-70.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9842	0,9848	0,9848	0,9852
$P(b\|B)$	0,9835	0,9842	0,9843	0,9851
$P(m\|M)$	0,9844	0,9852	0,9852	0,9858

Tabela 6.56: QDA 40-60.

A técnica QDA mostrou bons resultados, apresentando uma performance estável para as várias possibilidades de desbalanceamento dos dados. Como já é sabido, esses bons resultados para QDA é devido as suposições corretas feitas pelo modelo tais como, normalidade das variáveis explicativas e diferentes matrizes de variância entre as classes. Em um outro conjunto de dados possivelmente LDA e QDA teriam uma performance mais parecidas, ou não.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9843	0,9849	0,9849	0,9856
$P(b\|B)$	0,9801	0,9810	0,9810	0,9820
$P(m\|M)$	0,9880	0,9888	0,9888	0,9896

Tabela 6.57: QDA 50-50.

AUC	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
10%-90%	0,9985	0,9986	0,9986	0,9987
20%-80%	0,9985	0,9986	0,9986	0,9987
30%-70%	0,9985	0,9986	0,9986	0,9987
40%-60%	0,9985	0,9986	0,9986	0,9987
50%-50%	0,9985	0,9986	0,9986	0,9987

Tabela 6.58: Medidas AUC - QDA.

Em seguida, exibiremos os resultados obtidos com o modelo naive Bayes. Esse modelo é bastante conhecido da literatura e possui uma ideia muito simples como exibido no Capítulo 2. Os resultados obtidos com esse modelo foram bons, o motivo disso é que a distribuição de probabilidade utilizada pelo modelo foi a distribuião normal e os dados foram simulados também de uma distribuição normal. Além disso, esse modelo supõe independência condicional das variáveis explicativas o que é verdadeiro pois as matrizes de variâncias utilizadas para simular os dados possuem elementos fora da diagonal iguais a zero, conduzindo à independência entre as variáveis explicativas.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9711	0,9717	0,9718	0,9726
$P(b\|B)$	0,9932	0,9937	0,9937	0,9942
$P(m\|M)$	0,9486	0,9497	0,9500	0,9515

Tabela 6.59: naive Bayes 10-90.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9796	0,9803	0,9803	0,9810
$P(b\|B)$	0,9896	0,9901	0,9901	0,9907
$P(m\|M)$	0,9695	0,9705	0,9705	0,9715

Tabela 6.60: naive Bayes 20-80.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9829	0,9835	0,9835	0,9840
$P(b\|B)$	0,9867	0,9874	0,9875	0,9881
$P(m\|M)$	0,9785	0,9795	0,9795	0,9804

Tabela 6.61: naive Bayes 30-70.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9840	0,9846	0,9846	0,9854
$P(b\|B)$	0,9834	0,9841	0,9841	0,9849
$P(m\|M)$	0,9842	0,9852	0,9852	0,9861

Tabela 6.62: naive Bayes 40-60.

	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
ACC	0,9846	0,9852	0,9850	0,9856
$P(b\|B)$	0,9801	0,9812	0,9811	0,9822
$P(m\|M)$	0,9883	0,9890	0,9890	0,9896

Tabela 6.63: naive Bayes 50-50.

AUC	$1^{\underline{0}}$ Qu.	Mediana	Média	$3^{\underline{0}}$ Qu.
10%-90%	0,9985	0,9986	0,9986	0,9987
20%-80%	0,9985	0,9986	0,9986	0,9987
30%-70%	0,9985	0,9986	0,9986	0,9987
40%-60%	0,9985	0,9986	0,9986	0,9987
50%-50%	0,9985	0,9986	0,9986	0,9987

Tabela 6.64: Medidas AUC - naive Bayes.

Notamos através das Tabelas 6.59, 6.60, 6.61, 6.62, 6.63 e 6.64 que a técnica também teve um bom desempenho, apresentando boa estabilidade para as diferentes proporções de eventos e não eventos. Assim como no caso de QDA, possivelmente essa performance sofreria uma queda se as variáveis explicativas não fossem normais entretanto, com esses dados naive Bayes obteve boa performance.

Nas Tabelas 6.65, 6.66, 6.67 e 6.68 apresentam os valores médios para as medidas utilizadas nas comparações.

Técnica	10-90	20-80	30-70	40-60	50-50
Logística (0,3)	0,4983	0,4869	0,6439	0,6609	0,6081
Logística (0,4)	0,4999	0,4859	0,5125	0,643	0,6685
Logística (0,5)	0,5	0,4970	0,478	0,5306	0,6426
Rede Neural (3)	0,5	0,6078	0,7201	0,7619	0,7724
Rede Neural (6)	0,7029	0,801	0,8408	0,8582	0,861
Rede Neural (9)	0,7926	0,8629	0,8861	0,8958	0,8988
Árvore	0,7408	0,7879	0,8152	0,8329	0,8411
LDA	0,5	0,4978	0,4789	0,5279	0,6450
QDA	0,9712	0,98	0,9832	0,9848	0,9849
naive Bayes	0,9718	0,9803	0,9835	0,9846	0,985
SVM	0,9702	0,9799	0,9832	0,9835	0,9837

Tabela 6.65: Valores médios de ACC para as diferentes bases de treinamento.

Técnica	10-90	20-80	30-70	40-60	50-50
Logística (0,3)	$4 * 10^{-6}$	0,1011	0,6866	0,9521	0,9945
Logística (0,4)	0	0,003292	0,2056	0,684	0,9281
Logística (0,5)	0	$1,6 * 10^{-5}$	0,02147	0,2724	0,6821
Rede Neural (3)	0	0,2847	0,6077	0,7691	0,852
Rede Neural (6)	0,4375	0,6743	0,7891	0,8584	0,8996
Rede Neural (9)	0,6148	0,7775	0,8508	0,8940	0,9243
Árvore	0,4932	0,6032	0,6782	0,7403	0,7967
LDA	0	0	0,0171	0,2628	0,0171
QDA	0,9488	0,9701	0,9794	0,9852	0,9888
naive Bayes	0,95	0,9705	0,9795	0,9852	0,989
SVM	0,9476	0,9476	0,9793	0,9857	0,9895

Tabela 6.66: Valores médios de $P(m|M)$ para as diferentes bases de treinamento.

Técnica	10-90	20-80	30-70	40-60	50-50
Logística (0,3)	0,9965	0,8726	0,6012	0,3697	0,2216
Logística (0,4)	0,9997	0,9685	0,8193	0,602	0,409
Logística (0,5)	1	0,9941	0,9346	0,7888	0,6032
Rede Neural (3)	1	0,9309	0,8326	0,7547	0,6928
Rede Neural (6)	0,9683	0,9277	0,8926	0,8579	0,8223
Rede Neural (9)	0,9704	0,945	0,9215	0,8976	0,8733
Árvore	0,9883	0,9726	0,9522	0,9255	0,8855
LDA	1	0,9955	0,9406	0,7930	0,5977
QDA	0,9936	0,99	0,9871	0,9843	0,981
naive Bayes	0,9937	0,9901	0,9875	0,9841	0,9811
SVM	0,9928	0,9928	0,9853	0,9813	0,9779

Tabela 6.67: Valores médios de $P(b|B)$ para as diferentes bases de treinamento.

Técnica	10-90	20-80	30-70	40-60	50-50
Logística	0,672	0,6727	0,6724	0,6731	0,6728
Rede Neural (3)	0,8308	0,8287	0,8277	0,8298	0,8316
Rede Neural (6)	0,9255	0,9281	0,9298	0,9306	0,9289
Rede Neural (9)	0,9564	0,9599	0,9603	0,9598	0,9595
Árvore	0,8311	0,8435	0,8533	0,8612	0,8697
LDA	0,6725	0,6722	0,6728	0,6722	0,6723
QDA	0,9986	0,9986	0,9986	0,9986	0,9986
naive Bayes	0,9986	0,9986	0,9986	0,9986	0,9986
SVM	0,9983	0,9985	0,9984	0,9982	0,9981

Tabela 6.68: Valores médios de AUC para as diferentes bases de treinamento.

Observando as Tabelas, nota-se que SVM obteve bons resultados em todas as tabelas. As técnicas análise de discriminante quadrática e naive Bayes também apresentaram bons resultados, porém isso é fruto da normalidade dos dados. As técnicas rede neural e árvore obtiveram resultados bons para algumas configurações dos dados. Além disso, os melhores resultados para redes neurais foram obtidos para 9 elementos na camanda interna.

Com relação as Tabelas 6.69, 6.70, 6.71 e 6.72 o discurso é mesmo ao que foi apresentado nas tabelas de valores médios.

Um fenômeno interessante é que uma melhora em uma medida acarreta uma piora em outra, por exemplo, na medida $P(b|B)$ e $P(m|M)$. Quando ocorre uma melhora, outra piora e em alguns casos essa variação é mais evidente, como no caso da logística.

Técnica	10-90	20-80	30-70	40-60	50-50
Logística (0,3)	0,4983	0,4868	0,6438	0,6608	0,608
Logística (0,4)	0,4999	0,4858	0,5123	0,643	0,6689
Logística (0,5)	0,5	0,4971	0,4779	0,5308	06428
Rede Neural (3)	0,5	0,6163	0,7278	0,7645	0,7743
Rede Neural (6)	0,7017	0,805	0,8442	0,8603	0,8645
Rede Neural (9)	0,7897	0,8629	0,8868	0,8968	0,9002
Árvore	0,7406	0,7880	0,8149	0,8326	0,8408
LDA	0,5	0,4978	0,4788	0,5274	0,6445
QDA	0,9712	0,98	0,9832	0,9848	0,9849
naive Bayes	0,9717	0,9803	0,9835	0,9846	0,9852
SVM	0,97	0,98	0,982	0,9832	0,984

Tabela 6.69: Valores de mediana de ACC para as diferentes bases de treinamento.

Técnica	10-90	20-80	30-70	40-60	50-50
Logística (0,3)	0,9966	0,8730	0,6006	0,3698	0,2216
Logística (0,4)	0,9998	0,9686	0,8192	0,6015	0,4093
Logística (0,5)	1	0,9942	0,9346	0,789	0,6034
Rede Neural (3)	1	0,9253	0,8342	0,7604	0,6971
Rede Neural (6)	0,9681	0,9278	0,8934	0,8588	0,8259
Rede Neural (9)	0,9706	0,9451	0,9218	0,8980	0,8744
Árvore	0,9886	0,973	0,9524	0,926	0,8856
LDA	1	0,9956	0,9408	0,7936	0,5972
QDA	0,9936	0,99	0,987	0,9842	0,981
naive Bayes	0,9937	0,9901	0,9874	0,9841	0,9812
SVM	0,993	0,993	0,985	0,982	0,978

Tabela 6.70: Valores de mediana de $P(b|B)$ para as diferentes bases de treinamento.

As Figuras 6.1, 6.2, 6.3, 6.4 e 6.5 apresentam os valores das 4 medidas utilizadas para cada técnica, temporalmente por repetição. Esses gráficos auxiliam na verificação de estabilidade das técnicas e também na construção de uma relação de desempenho preditio entre as mesmas. Vemos, por exemplo, que a rede neural com 3 elementos na camada interna algumas vezes apresenta uma queda na performance, constatamos isso através das quedas abruptas apresentadas nas imagens.

Técnica	10-90	20-80	30-70	40-60	50-50
Logística (0,3)	0	0,101	0,6862	0,9523	0,9945
Logística (0,4)	0	0,0032	02048	0,6846	0,9282
Logística (0,5)	0	0	0,02145	0,2724	0,6821
Rede Neural (3)	0	0,3092	0,6206	0,7688	0,8518
Rede Neural (6)	0,4358	0,684	0,7947	0,8615	0,9022
Rede Neural (9)	0,6108	0,7802	0,8448	0,8948	0,9247
Árvore	0,4934	0,6044	0,6767	0,74	07961
LDA	0	0	0,01725	0,2622	0,01725
QDA	0,9486	0,97	0,9793	0,9852	0,9888
naive Bayes	0,9497	0,9705	0,9795	0,9852	0,989
SVM	0,948	0,948	0,9795	0,986	0,99

Tabela 6.71: Valores de mediana de $P(m|M)$ para as diferentes bases de treinamento.

Técnica	10-90	20-80	30-70	40-60	50-50
Logística	0,6722	0,6721	0,6717	0,6729	0,673
Rede Neural (3)	0,8327	0,8333	0,8334	0,8334	0,8347
Rede Neural (6)	0,9252	0,9307	0,932	0,932	0,9317
Rede Neural (9)	0,9558	0,9605	0,9607	0,9604	0,9604
Árvore	0,8316	0,8432	0,8534	0,8612	0,8698
LDA	0,6725	0,6719	0,6728	0,6722	0,6723
QDA	0,9986	0,9986	0,9986	0,9986	0,9986
naive Bayes	0,9986	0,9986	0,9986	0,9986	0,9986
SVM	0,9984	0,9985	0,9985	0,9983	0,9981

Tabela 6.72: Valores de medianas de AUC para as diferentes bases de treinamento.

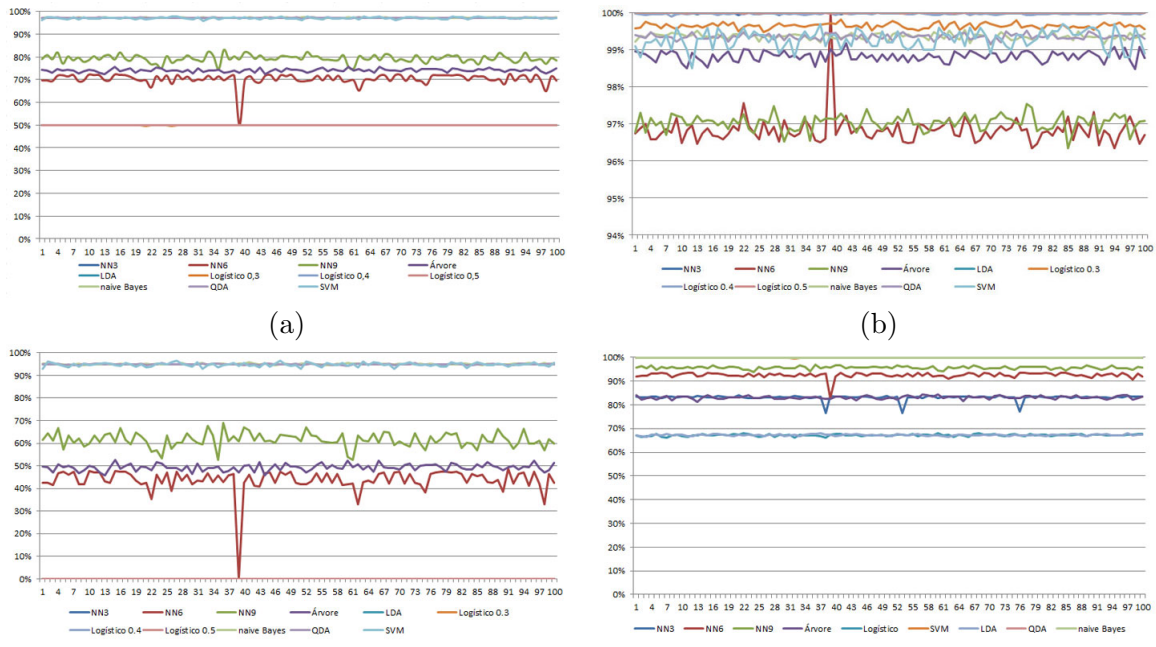


Figura 6.1: (a) ACC; (b) $P(b|B)$; (c) $P(m|M)$; (d) AUC, medidas obtidas com 10% de maus e 90% de bons

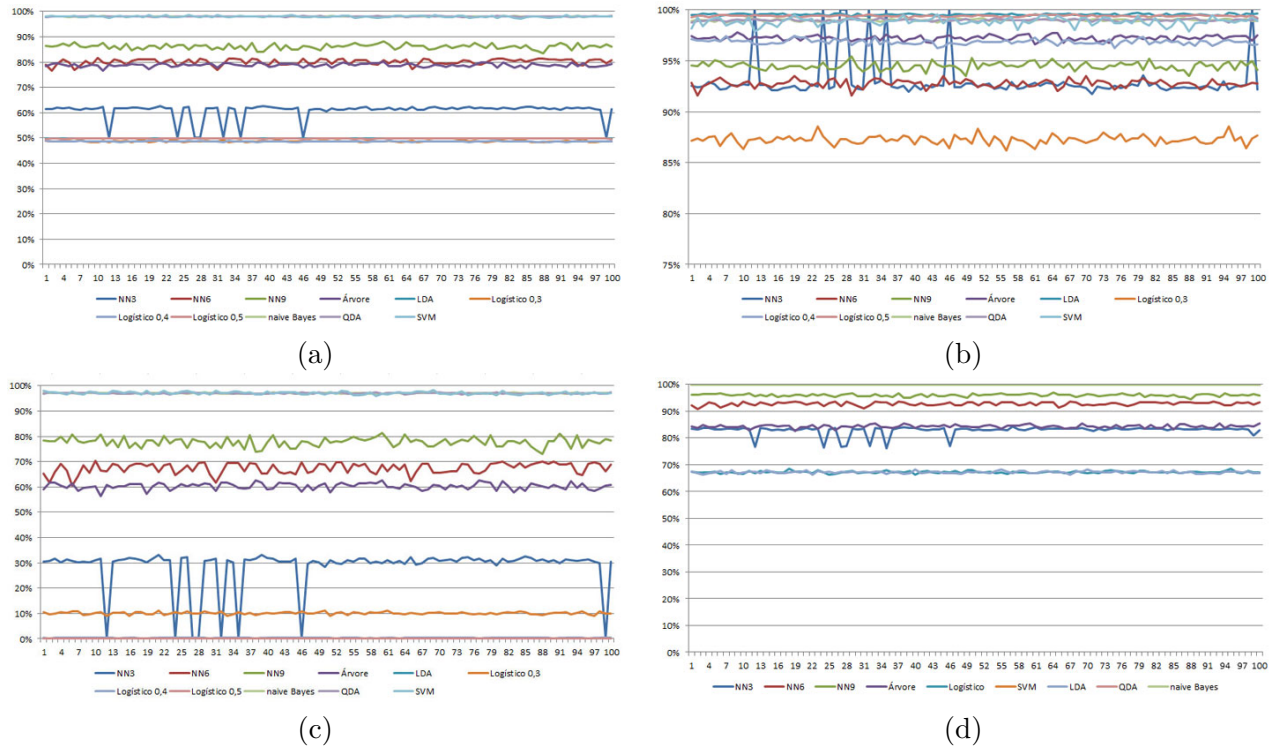


Figura 6.2: (a) ACC; (b) $P(b|B)$; (c) $P(m|M)$; (d) AUC, medidas obtidas com 20% de maus e 80% de bons

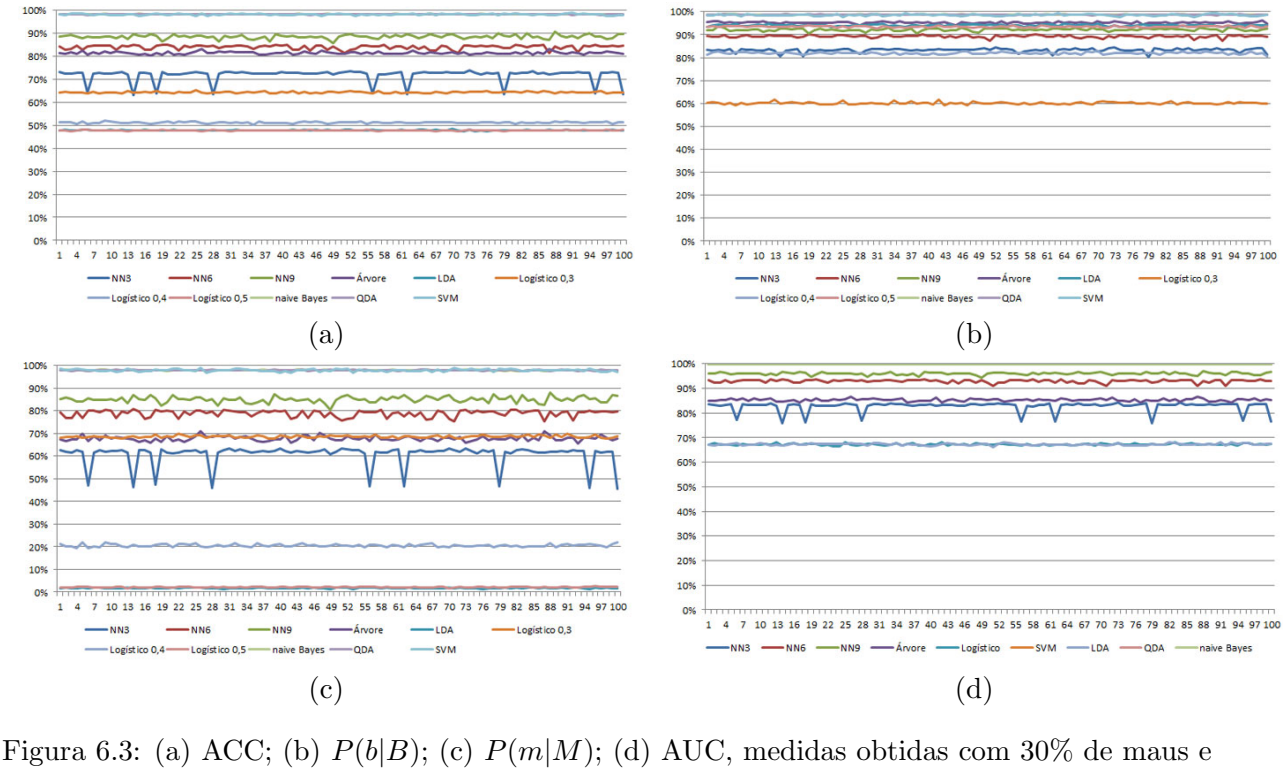


Figura 6.3: (a) ACC; (b) $P(b|B)$; (c) $P(m|M)$; (d) AUC, medidas obtidas com 30% de maus e 70% de bons

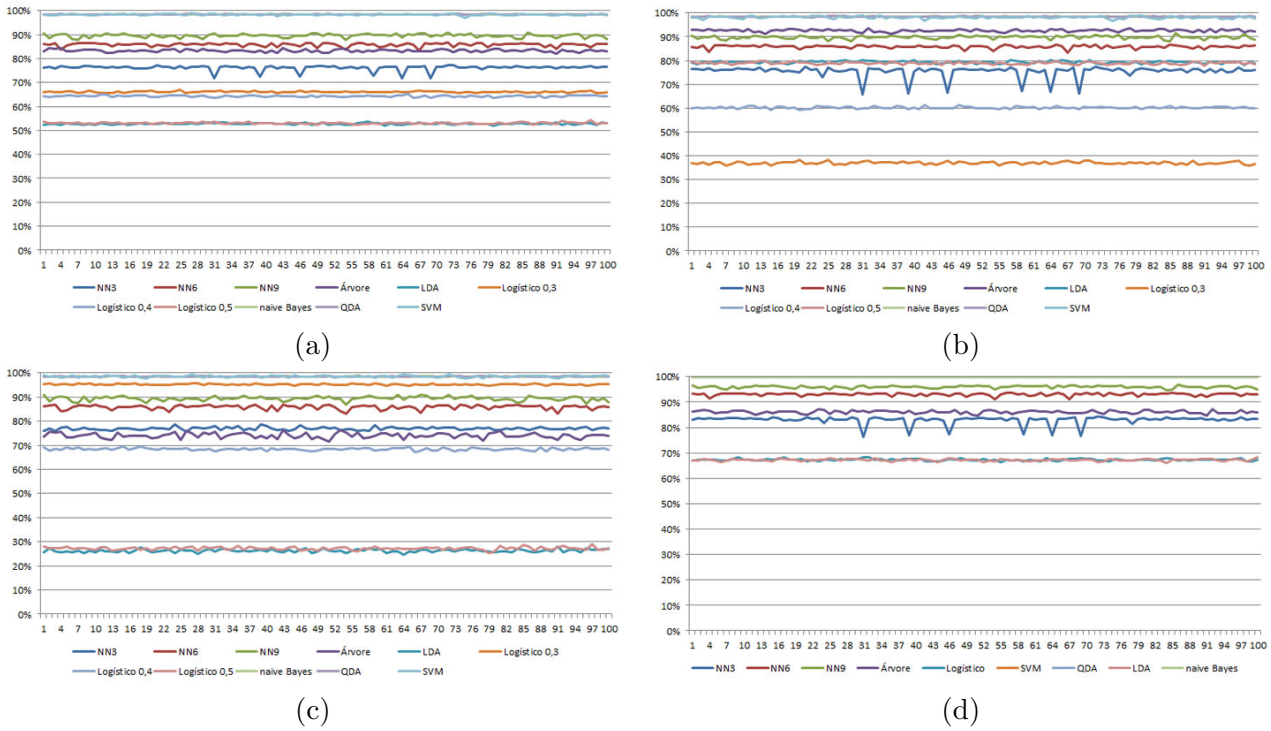


Figura 6.4: (a) ACC; (b) $P(b|B)$; (c) $P(m|M)$; (d) AUC, medidas obtidas com 40% de maus e 60% de bons

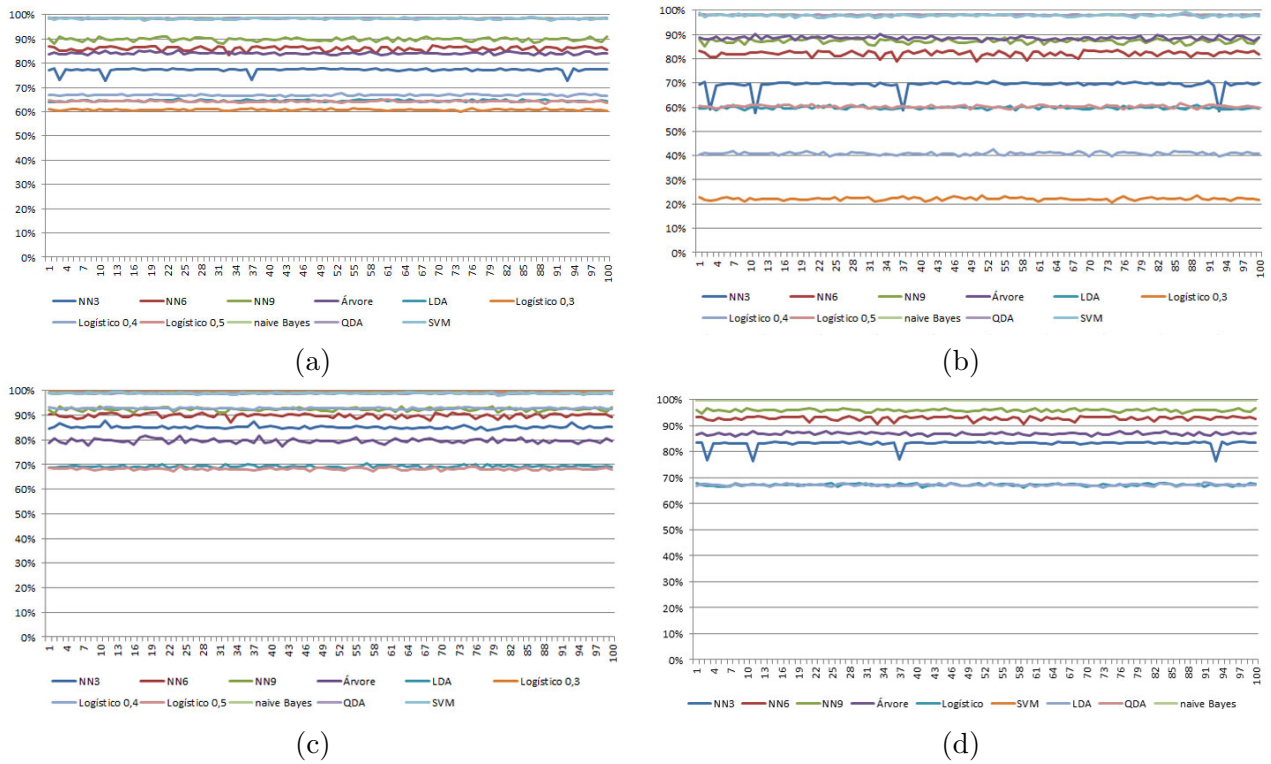


Figura 6.5: (a) ACC; (b) $P(b|B)$; (c) $P(m|M)$; (d) AUC, medidas obtidas com 50% de maus e 50% de bons

6.2 Comentários Finais

Neste capítulo ralizamos um estudo de simulação para avaliar a performance de várias técnicas comumente utilizadas não somente em problemas de *credit scoring* mas também em problemas de classificação binária em geral. Algumas técnicas se mostraram melhor com relação a capacidade preditiva e melhor estabilidade. Houve casos de técnicas que se ajustaram bem ao problema de devido a normalidade dos dados.

Capítulo 7

CONCLUSÃO

Neste trabalho realizamos uma revisão das técnicas mais utilizadas em *credit scoring*, comentamos cada uma delas e descrevemos a ideia geral que as motiva. Foi possível constatar a grande variedade de técnicas cada uma com suas suposiçoes acerca dos dados como, por exemplo, a suposição de distribuições de densidade/probabilidade dos dados e suposições sobre a variância dos dados.

Realizamos, também, uma revisão sobre as medidas de performance mais utilizadas, expondo algumas metodologias, vantagens e desvantagens. Em seguida, um levantamento das técnicas mais utilizadas e sua variação ao longo do tempo. Além disso, foi feita uma comparação mais detalhada para as três técnicas mais utilizadas conforme o estudo.

O estudo de simulação envolveu várias técnicas entre elas as três mais utilizadas, segundo o estudo de revisão bibliografica sistemática. Além disso, a simulação foi realizada com uma base de dados artificiais em que a variável resposta é binária e as covariáveis são normais multivariadas e com diferentes proporções entre eventos e não eventos. Calculamos as medidas preditivas para cada técnica e os valores são exibidos em Tabelas com os quantis para obtermos uma noção da variabilidade das medidas obtidas. Para que fosse possível fazer uma comparação direta das técnicas, foram construídas Tabelas com os valores de performance para cada técnica. Além disso, as técnicas foram comparadas através de gráficos com as medidas de performance

A técnica SVM mostrou-se bastante estável nos vários cenários possíveis, sendo portanto uma boa opção a ser considerada. Entretanto, com base nesse estudo não é possível excluir todas as outras técnicas uma vez que existe existem vários tipos e estruturas de conjuntos de dados reais e/ou simulados e deve-se fazer um estudo com vários modelos antes de tomar uma decisão de qual método é o mais adequado.

Desta forma, esse texto não esgota todas as possibilidades pois, é possível fazer um estudo de simulação mais amplo e que englobe conjuntos de dados mais distintos. Além disso, é possível utilizar duas ou mais técnicas diferentes para tentar obter melhor performance preditiva, criando

assim uma técnica híbrida.

Uma possível continuação para este trabalho é a realização de uma revisão bibliografica sistemática e de estudos comparativos entre as diferentes possibilidades de construção de técnicas híbridas.

Referências Bibliográficas

[1] Abdou, H. Genetic programming for credit scoring: The case of Egyptian public sector banks. Expert Systems with Applications, 2009, Vol. 36(9), pp. 11402-11417

[2] Abdou, H., Pointon, J. and El-Masry, A. Neural nets versus conventional techniques in credit scoring in Egyptian banking. Expert Systems with Applications, 2008, Vol. 35(3), pp. 1275-1292

[3] Abdoun, O. and Abouchabaka, J. A Comparative Study of Adaptive Crossover Operators for Genetic Algorithms to Resolve the Traveling Salesman Problem. International Journal of Computer Applications, 2011, Vol. 31(11), pp. 49-57

[4] Adams, N., Hand, D. and Till, R. Mining for classes and patterns in behavioural data. Journal of the Operational Research Society, 2001, Vol. 52(9), pp. 1017-1024

[5] Akkoc, S. An empirical comparison of conventional techniques, neural networks and the three stage hybrid Adaptive Neuro Fuzzy Inference System (ANFIS) model for credit scoring analysis: The case of Turkish credit card data. European Journal of Operational Research, 2012, Vol. 222(1), pp. 168-178

[6] Antonakis, A. and Sfakianakis, M. Assessing naïve Bayes as a method for screening credit applicants. Journal of Applied Statistics, 2009, Vol. 36(5), pp. 537-545

[7] Aussem, A., Tchernof, A., de Morais, S. R. and Rome, S. Analysis of lifestyle and metabolic predictors of visceral obesity with Bayesian Networks. BCM Bioinformatics, 2010, Vol. 11, pp. 487

[8] Bache, K. and Lichman, M. UCI Machine Learning Repository. University of California, Irvine, School of Information and Computer Sciences, 2013

[9] Baesens, B., Van Gestel, T., Stepanova, M., Van Den Poel, D. and Vanthienen, J. Neural network survival analysis for personal loan data. Journal of the Operational Research Society, 2005, Vol. 56(9), pp. 1089-1098

[10] Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J. and Vanthienen, J. Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society, 2003, Vol. 54(6), pp. 627-635

[11] Baldi, P., Brunak, S., Chauvin, Y., Andersen, C. A. F. and Nielsen, H. Assessing the accuracy of prediction algorithms for classification: an overview Bioinformatics Review, 2000, Vol. 16(5), pp. 412-424

[12] Banasik, J., Crook, J. and Thomas, L. Sample selection bias in credit scoring models. Journal of the Operational Research Society, 2003, Vol. 54(8), pp. 822-832

[13] Banasik, J., Crook, J. and Thomas, L. Not if but when will borrowers default. Journal of the Operational Research Society, 1999, Vol. 50(12), pp. 1185-1190

[14] Bardos, M. Detecting the risk of company failure at the Banque de France. Journal of Banking and Finance, 1998, Vol. 22(10-11), pp. 1405-1419

[15] Baxter, R., Gawler, M. and Ang, R. Predictive model of insolvency risk for australian corporations. Conferences in Research and Practice in Information Technology Series, 2007, Vol. 70, pp. 21-28

[16] Becker, U. and Fahrmeir, L. Bump Hunting for Risk: A New Data Mining Tool and its Applications. Computational Statistics, 2001, Vol. 16(3), pp. 373-386

[17] Ben-David, A. and Frank, E. Accuracy of machine learning models versus hand craft. Expert Systems with Applications, 2009, Vol. 36(3 PART 1), pp. 5264-5271

[18] Berger, A., Frame, W. and Miller, N. Credit scoring and the availability, price, and risk of small business credit. Journal of Money, Credit and Banking, 2005, Vol. 37(2), pp. 191-222

[19] Berkson, J. Application of the logistic function to bio-assay. Journal of the American Statistical Association, Taylor & ; Francis, 1944, Vol. 39(227), pp. 357-365

[20] Berry, M. J. A. and Linoff, G. Data mining techniques. USA: John Wiley, 1997

[21] Bijak, K. and Thomas, L. Does segmentation always improve model performance in credit scoring? Expert Systems with Applications, 2012, Vol. 39(3), pp. 2433-2442

[22] Bishop, C. M. Neural networks for pattern recognition. Oxford university press, 1995

[23] Bliss, C. I. The calculation of the dosage-mortality curve. Annals of Applied Biology, 1935, Vol. 22, pp. 134-167

[24] Bobbio, A., Portinale, L., Minichino, M. and Ciancarmela, E. Improving the Analysis of Dependable Systems by Mapping Fault Trees into Bayesian Networks. Realiability Engineering &; System Safety, 2001, Vol. 71, pp. 249-260

[25] Breiman, L. Arcing classifiers. The Annals of Statistics, 1998, Vol. 26, pp. 801 - 849.

[26] Breiman, L. Bagging predictors. Machine learning, Springer, 1996, Vol. 24(2), pp. 123-140

[27] Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. Classification and regression trees. Wadsworth &; Brooks Monterey, CA, 1984

[28] Brown, I. and Mues, C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Systems with Applications, 2012, Vol. 39(3), pp. 3446-3453

[29] Burset, M. and Guig, R. Evaluation of gene structure prediction programs. Genomics, 1996, Vol. 34, pp. 353-367

[30] Burton, D. Credit scoring, risk, and consumer lendingscapes in emerging markets. Environment and Planning A, 2012, Vol. 44(1), pp. 111-124

[31] Capotorti, A. and Barbanera, E. Credit scoring analysis using a fuzzy probabilistic rough set model. Computational Statistics and Data Analysis, 2012, Vol. 56(4), pp. 981-994

[32] Chang, K. C., Fung, R., Lucas, A., Oliver, R. and Shikaloff, N. Bayesian networks applied to credit scoring. IMA Journal of Mathematics Applied in Business and Industry, 2000, Vol. 11, pp. 1-18

[33] Chang, S.-Y. and Yeh, T.-Y. An artificial immune classifier for credit scoring analysis. Applied Soft Computing Journal, 2012, Vol. 12(2), pp. 611-618

[34] Chen, F.-L. and Li, F.-C. Comparison of the hybrid credit scoring models based on various classifiers. International Journal of Intelligent Information Technologies, 2010, Vol. 6(3), pp. 56-74

[35] Chen, F.-L. and Li, F.-C. Combination of feature selection approaches with SVM in credit scoring. Expert Systems with Applications, 2010, Vol. 37(7), pp. 4902-4909

[36] Chen, G. and Astebro, T. Bound and collapse Bayesian reject inference for credit scoring. Journal of the Operational Research Society, 2012, Vol. 63(10), pp. 1374-1387

[37] Chen, M.-C. and Huang, S.-H. Credit scoring and rejected instances reassigning through evolutionary computation techniques. Expert Systems with Applications, 2003, Vol. 24(4), pp. 433-441

[38] Chen, W., Ma, C. and Ma, L. Mining the customer credit using hybrid support vector machine technique. Expert Systems with Applications, 2009, Vol. 36(4), pp. 7611-7616

[39] Cheng, J., Bell, D. A. and Liu, W. An algorithm for Bayesian network construction from data. Proceedings of the 6th International Workshop on Artificial Intelligence and Statistics 1997

[40] Chrzanowska, M., Alfaro, E. and Witkowska, D. The individual borrowers recognition: Single and ensemble trees. Expert Systems with Applications, 2009, Vol. 36(3 PART 2), pp. 6409-6414

[41] Chuang, C.-L. and Huang, S.-T. A hybrid neural network approach for credit scoring. Expert Systems, 2011, Vol. 28(2), pp. 185-196

[42] Chuang, C.-L. and Lin, R.-H. Constructing a reassigning credit scoring model. Expert Systems with Applications, 2009, Vol. 36(2 PART 1), pp. 1685-1694

[43] Constangioara, A. and Marton, O. Statistical instruments used in credit scoring. Journal of Electrical and Electronics Engineering, 2010, Vol. 3(2), pp. 55-58

[44] DeYoung, R., Frame, W., Glennon, D. and Nigro, P. The Information Revolution and Small Business Lending: The Missing Evidence. Journal of Financial Services Research, 2011, Vol. 39(41306), pp. 19-33

[45] Dong, Y., Hao, X. and Yu, C. Comparison of statistical and artificial intelligence methodologies in small-businesses' credit assessment based on daily transaction data. ICIC Express Letters, 2011, Vol. 5(5), pp. 1725-1730

[46] Dryver, A. and Sukkasem, J. Validating risk models with a focus on credit scoring models. Journal of Statistical Computation and Simulation, 2009, Vol. 79(2), pp. 181-193

[47] Durand, D. Risk elements in consumer instalment financing. National Bureau of Economics, 1941

[48] Efromovich, S. Oracle inequality for conditional density estimation and an actuarial example. Annals of the Institute of Statistical Mathematics, 2010, Vol. 62(2), pp. 249-275

[49] Ewout W. Steyerberg Andrew J. Vickers, N. R. C. T. G. M. G. N. O. M. J. P. and Kattan, M. W. Assessing the performance of prediction models: a framework for some traditional and novel measures. NIH Public Access, 2010, Vol. 21(1), pp. 128-138

[50] Falangis, K. and Glen, J. Heuristics for feature selection in mathematical programming discriminant analysis models. Journal of the Operational Research Society, 2010, Vol. 61(5), pp. 804-812

[51] Feng, L., Yao, Y. and Jin, B. Research on credit scoring model with SVM for network management. Journal of Computational Information Systems, 2010, Vol. 6(11), pp. 3567-3574

[52] Fernandez, A. and Salmeron, A. BayesChess: A computer chess program based on Bayesian networks. Pattern Recognition Letters, 2008, Vol. 29(8), pp. 1154-1159

[53] Figini, S. and Uberti, P. Model assessment for predictive classification models. Communications in Statistics - Theory and Methods, 2010, Vol. 39(18), pp. 3238-3244

[54] Finlay, S. Multiple classifier architectures and their application to credit risk assessment. European Journal of Operational Research, 2011, Vol. 210(2), pp. 368-378

[55] Finlay, S. Credit scoring for profitability objectives. European Journal of Operational Research, 2010, Vol. 202(2), pp. 528-537

[56] Finlay, S. Are we modelling the right thing? The impact of incorrect problem specification in credit scoring. Expert Systems with Applications, 2009, Vol. 36(5), pp. 9065-9071

[57] Finlay, S. Towards profitability: A utility approach to the credit scoring problem. Journal of the Operational Research Society, 2008, Vol. 59(7), pp. 921-931

[58] Fisher, R. A. The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics, 1986, Vol. 7, pp. 179-188

[59] Friedman, N., Geiger, D. and Goldszmidt, M. Bayesian network classifiers. Machine Learning, 1997, Vol. 29(2-3), pp. 131-163

[60] Gemela, J. Financial analysis using Bayesian networks. Applied Stochastic Models in Business and Industry, 2001, Vol. 17(1), pp. 57-67

[61] Gestel, T., Baesens, B., Suykens, J., Van den Poel, D., Baestaens, D.-E. and Willekens, M. Bayesian kernel based classification for financial distress detection. European Journal of Operational Research, 2006, Vol. 172(3), pp. 979-1003

[62] Giudici, P. Bayesian data mining, with application to benchmarking and credit scoring. Applied Stochastic Models in Business and Industry, 2001, Vol. 17(1), pp. 69-81

[63] Goletsis, Y., Exarchosb, T. and Katsis, C. Credit scoring using an Ant mining approach. Human Systems Management, 2010, Vol. 29(2), pp. 79-88

[64] Goodman, L. A. and Kruskal, W. H. Measures of association for cross classifications. Part III. J. Amer. Statist. Assoc., 1963, Vol. 58, pp. 310-364

[65] Hachicha, W. and Ghorbel, A. A survey of control-chart pattern-recognition literature (1991–2010) based on a new conceptual classification scheme. Computers &; Industrial Engineering, Elsevier, 2012, Vol. 63(1), pp. 204-222

[66] Hamilton, D., Riley, P. J., Miola, U. J. and Amro, A. A. A feed forward neural network for classification of bulls eye myocardial perfusion images. European Journal of Nuclear Medicine, 1995, Vol. 22, pp. 108-115

[67] Han, J., Kamber, M. and Pei, J. Data mining: concepts and techniques. Morgan kaufmann, 2006

[68] Hand, D. Supervised classification and tunnel vision Applied Stochastic Models in Business and Industry, 2005, Vol. 21(2), pp. 97-109

[69] Hand, D. Good practice in retail credit scorecard assessment. Journal of the Operational Research Society, 2005, Vol. 56(9), pp. 1109-1117

[70] Hand, D. Modelling consumer credit risk. IMA Journal Management Mathematics, 2001, Vol. 12(2), pp. 139-155

[71] Hand, D. and Henley, W. Statistical classification methods in consumer credit scoring: A review Journal of the Royal Statistical Society. Series A: Statistics in Society, 1997, Vol. 160(3), pp. 523-541

[72] Hand, D. and Kelly, M. Superscorecards. IMA Journal Management Mathematics, 2002, Vol. 13(4), pp. 273-281

[73] Hand, D. and Kelly M.G. Lookahead scorecards for new fixed term credit products Journal of the Operational Research Society, 2001, Vol. 52(9), pp. 989-996

[74] Hand, D. J. Discrimination and Classification. New York: John Wiley &; Sons, Inc, 1981

[75] Hand D.J. Good practice in retail credit scorecard assessment Journal of the Operational Research Society, 2005, Vol. 56(9), pp. 1109-1117

[76] Hand D.J. Supervised classification and tunnel vision. Applied Stochastic Models in Business and Industry, 2005, Vol. 21(2), pp. 97-109

[77] Hand D.J. Modelling consumer credit risk. IMA Journal Management Mathematics, 2001, Vol. 12(2), pp. 139-155

[78] Hardle, W., Mammen, E. and Muller, M. Testing parametric versus semiparametric modeling in generalized linear models. Journal of the American Statistical Association, 1998, Vol. 93(444), pp. 1461-1474

[79] He, J., Zhang, Y., Shi, Y. and Huang, G. Domain-driven classification based on multiple criteria and multiple constraint-level programming for intelligent credit scoring. IEEE Transactions on Knowledge and Data Engineering, 2010, Vol. 22(6), pp. 826-838

[80] Hens, A. and Tiwari, M. Computational time reduction for credit scoring: An integrated approach based on support vector machine and stratified sampling method. Expert Systems with Applications, 2012, Vol. 39(8), pp. 6774-6781

[81] Hoffmann, F., Baesens, B., Martens, J., Put, F. and Vanthienen, J. Comparing a genetic fuzzy and a neurofuzzy classifier for credit scoring. International Journal of Intelligent Systems, 2002, Vol. 17(11), pp. 1067-1083

[82] Hoffmann, F., Baesens, B., Mues, C., Van Gestel, T. and Vanthienen, J. Inferring descriptive and approximate fuzzy rules for credit scoring using evolutionary algorithms. European Journal of Operational Research, 2007, Vol. 177(1), pp. 540-555

[83] Hosmer, W. D. and Lemeshow, S. Applied Logistic Regression. Wiley, 1989

[84] Hsieh, N.-C. Hybrid mining approach in the design of credit scoring models. Expert Systems with Applications, 2005, Vol. 28(4), pp. 655-665

[85] Hsieh, N.-C. and Hung, L.-P. A data driven ensemble classifier for credit scoring analysis. Expert Systems with Applications, 2010, Vol. 37(1), pp. 534-545

[86] Hu, Y.-C. and Ansell, J. Retail default prediction by using sequential minimal optimization technique. Journal of Forecasting, 2009, Vol. 28(8), pp. 651-666

[87] Hu, Y.-C. and Ansell, J. Measuring retail company performance using credit scoring techniques. European Journal of Operational Research, 2007, Vol. 183(3), pp. 1595-1606

[88] Huang, C.-L., Chen, M.-C. and Wang, C.-J. Credit scoring with a data mining approach based on support vector machines. Expert Systems with Applications, 2007, Vol. 33(4), pp. 847-856

[89] Huang, J.-J., Tzeng, G.-H. and Ong, C.-S. Two-stage genetic programming (2SGP) for the credit scoring model. Applied Mathematics and Computation, 2006, Vol. 174(2), pp. 1039-1053

[90] Huang, Y.-M., Hung, C.-M. and Jiau, H. Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem. Nonlinear Analysis: Real World Applications, 2006, Vol. 7(4), pp. 720-747

[91] Huysmans, J., Baesens, B., Vanthienen, J. and Van Gestel, T. Failure prediction with self organizing maps. Expert Systems with Applications, 2006, Vol. 30(3), pp. 479-487

[92] Ince, H. and Aktan, B. A comparison of data mining techniques for credit scoring in banking: A managerial perspective. Journal of Business Economics and Management, 2009, Vol. 10(3), pp. 233-240

[93] Jin, X., Zhou, W. and Bie, R. Multinomial event naive Bayesian modeling for SAGE data classification. Computational Statistics, 2007, Vol. 22, pp. 133-143

[94] John, G., Miller, P. and Kerber, R. Stock selection using rule induction. IEEE Expert-Intelligent Systems and their Applications, 1996, Vol. 11(5), pp. 52-58

[95] Jung, K. and Thomas, L. A note on coarse classifying in acceptance scorecards. Journal of the Operational Research Society, 2008, Vol. 59(5), pp. 714-718

[96] Kao, L.-J., Chiu, C.-C. and Chiu, F.-Y. A Bayesian latent variable model with classification and regression tree approach for behavior and credit scoring. Knowledge-Based Systems, 2012, Vol. 36, pp. 245-252

[97] Karlis, D. and Rahmouni, M. Analysis of defaulters' behaviour using the Poisson-mixture approach. IMA Journal Management Mathematics, 2007, Vol. 18(3), pp. 297-311

[98] Kiang, M. Y. A comparative assessment of classification methods. Decision Support Systems, 2003, Vol. 35, pp. 441-454

[99] Kocenda, E. and Vojtek, M. Default predictors in retail credit scoring: Evidence from Czech banking data. Emerging Markets Finance and Trade, 2011, Vol. 47(6), pp. 80-98

[100] Kolbe, R. and Brunette, M. Content analysis research: An examination of applications with directives for improving research, reliability and objectivity. Journal of Consumer Research, 1991, Vol. 18(2), pp. 243-250

[101] Korb, K. B. and Nicholson, A. E. Bayesian artificial intelligence. London: Chapman &; HallCRC Press UK, 2004

[102] Koza, J. R. Genetic programming: on the programming of computers by means of natural selection (complex adaptive systems) ¡dd¿The MIT Press Cambridge, 1992

[103] Laha, A. Building contextual classifiers by integrating fuzzy rule based classification technique and k-nn method for credit scoring. Advanced Engineering Informatics, 2007, Vol. 21(3), pp. 281-291

[104] Lahsasna, A., Ainon, R. and Wah, T. Enhancement of transparency and accuracy of credit scoring models through genetic fuzzy classifier. Maejo International Journal of Science and Technology, 2010, Vol. 4(1), pp. 136-158

[105] Lahsasna, A., Ainon, R. and Wah, T. Credit scoring models using soft computing methods: A survey International Arab Journal of Information Technology, 2010, Vol. 7(2), pp. 115-123

[106] Lan, Y., Janssens, D., Chen, G. and Wets, G. Improving associative classification by incorporating novel interestingness measures. Expert Systems with Applications, 2006, Vol. 31(1), pp. 184-192

[107] Lauria, E. J. and Duchessi, P. J. A Bayesian Belief Network for IT implementation decision support. Decision Support Systems, 2006, Vol. 42, pp. 1573-1588

[108] Lee, T.-S. and Chen, I.-F. A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. Expert Systems with Applications, 2005, Vol. 28(4), pp. 743-752

[109] Lee, T.-S., Chiu, C.-C., Chou, Y.-C. and Lu, C.-J. Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. Computational Statistics and Data Analysis, 2006, Vol. 50(4), pp. 1113-1130

[110] Lee, T.-S., Chiu, C.-C., Lu, C.-J. and Chen, I.-F. Credit scoring using the hybrid neural discriminant technique. Expert Systems with Applications, 2002, Vol. 23(3), pp. 245-254

[111] Leung, K., Cheong, F., Cheong, C., O'Farrell, S. and Tissington, R. Application of variable selection techniques to a modified SAIS for generating practical scorecards. International Journal of Applied Decision Sciences, 2009, Vol. 2(3), pp. 233-261

[112] Li, H. and Hand, D. Direct versus indirect credit scoring classifications. Journal of the Operational Research Society, 2002, Vol. 53(6), pp. 647-654

[113] Li, S.-T., Shiue, W. and Huang, M.-H. The evaluation of consumer loans using support vector machines Expert Systems with Applications, 2006, Vol. 30(4), pp. 772-782

[114] Li, T. and Cavusgil, S. T. A classification and assessment of research streams in international marketing. International Business Review, 1995, Vol. 4(3), pp. 251-277

[115] Li, X.-S., Guo, C.-X. and Guo, Y.-H. The credit scoring model on extended tree augment naive Bayesian network. Xitong Gongcheng Lilun yu ShijianSystem Engineering Theory and Practice, 2008, Vol. 28(6), pp. 129-136

[116] Lin, S.-M., Ansell, J. and Andreeva, G. Predicting default of a small business using different definitions of financial distress. Journal of the Operational Research Society, 2012, Vol. 63(4), pp. 539-548

[117] Ling, Y., Cao, Q. and Zhang, H. Credit scoring using multi-kernel support vector machine and chaos particle swarm optimization. International Journal of Computational Intelligence and Applications, 2012, Vol. 11(3), pp. 12500198:1-12500198:13

[118] Lisboa, P., Etchells, T., Jarman, I., Arsene, C., Aung, M., Eleuteri, A., Taktak, A., Ambrogi, F., Boracchi, P. and Biganzoli, E. Partial logistic artificial neural network for competing risks regularized with automatic relevance determination. IEEE Transactions on Neural Networks, 2009, Vol. 20(9), pp. 1403-1416

[119] Little, R. J. and Rubin, D. B. Statistical analysis with missing data. Wiley, 2002

[120] Liu, X., Fu, H. and Lin, W. A modified support vector machine model for credit scoring. International Journal of Computational Intelligence Systems, 2010, Vol. 3(6), pp. 797-803

[121] Liu, Y. and Schumann, M. Data mining feature selection for credit scoring models. Journal of the Operational Research Society, 2005, Vol. 56(9), pp. 1099-1108

[122] Louzada, F., Anacleto-Junior, O., Candolo, C. and Mazucheli, J. Poly-bagging predictors for classification modelling for credit scoring. Expert Systems with Applications, 2011, Vol. 38(10), pp. 12717-12720

[123] Louzada, F., Cancho, V., Roman, M. and Leite, J. A new long-term lifetime distribution induced by a latent complementary risk framework. Journal of Applied Statistics, 2012, Vol. 39(10), pp. 2209-2222

[124] Louzada, F., Ferreira-Silva, P. and Diniz, C. On the impact of disproportional samples in credit scoring models: An application to a Brazilian bank data. Expert Systems with Applications, 2012, Vol. 39(9), pp. 8071-8078

[125] Lu, A.-G., Wang, J. and Liu, H.-W. An improved SVM learning algorithm and its applications to credit scorings. Xitong Gongcheng Lilun yu ShijianSystem Engineering Theory and Practice, 2012, Vol. 32(3), pp. 515-521

[126] Lucas, A. Statistical challenges in credit card issuing. Applied Stochastic Models in Business and Industry, 2001, Vol. 17(1), pp. 69-81

[127] Lugovskaya, L. Predicting default of Russian SMEs on the basis of financial and non-financial variables. Journal of Financial Services Marketing, 2010, Vol. 14(4), pp. 301-313

[128] Luo, S.-T., Cheng, B.-W. and Hsieh, C.-H. Prediction model building with clustering-launched classification and support vector machines in credit scoring. Expert Systems with Applications, 2009, Vol. 36(4), pp. 7562-7566

[129] Malik, M. and Thomas L.C. Transition matrix models of consumer credit ratings. International Journal of Forecasting, 2012, Vol. 28(1), pp. 261-272

[130] Marcano-Cedeno, A., Marin-De-La-Barcena, A., Jimenez-Trillo, J., Pinuela, J. and Andina, D. Artificial metaplasticity neural network applied to credit scoring. International Journal of Neural Systems, 2011, Vol. 21(4), pp. 311-317

[131] Marques, A., Garcia, V. and Sanchez, J. Exploring the behaviour of base classifiers in credit scoring ensembles. Expert Systems with Applications, 2012, Vol. 39(11), pp. 10244-10250

[132] Marques, A., Garcia, V. and Sanchez, J. Two-level classifier ensembles for credit risk assessment. Expert Systems with Applications, 2012, Vol. 39(12), pp. 10916-10922

[133] Marron, D. 'Lending by numbers': Credit scoring and the constitution of risk within American consumer credit. Economy and Society, 2007, Vol. 36(1), pp. 103-133

[134] Martens, D., Baesens, B., Van Gestel, T. and Vanthienen, J. Comprehensible credit scoring models using rule extraction from support vector machines. European Journal of Operational Research, 2007, Vol. 183(3), pp. 1466-1476

[135] Martens, D., Van Gestel, T., De Backer, M., Haesen, R., Vanthienen, J. and Baesens, B. Credit rating prediction using Ant Colony Optimization. Journal of the Operational Research Society, 2010, Vol. 61(4), pp. 561-573

[136] Maznevski, M., Kemp, R., Overstreet, G. and Crook, J. The power to borrow and lend: investigating the cultural context as part of the lending decision. Journal of the Operational Research Society, 2001, Vol. 52(9), pp. 1045-1056

[137] McDonald, R., Sturgess, M., Smith, K., Hawkins, M. and Huang, E. Non-linearity of scorecard log-odds. International Journal of Forecasting, 2012, Vol. 28(1), pp. 239-247

[138] Mitchell, T. M. Machine learning. 1997. Burr Ridge, IL: McGraw Hill, 1997, Vol. 45

[139] Moon, T. and Sohn, S. Survival analysis for technology credit scoring adjusting total perception. Journal of the Operational Research Society, 2011, Vol. 62(6), pp. 1159-1168

[140] Mues, C., Baesens, B., Files, C. and Vanthienen, J. Decision diagrams in machine learning: An empirical study on real-life credit-risk data. Expert Systems with Applications, 2004, Vol. 27(2), pp. 257-264

[141] Neapolitan, R. E. Learning Bayesian Networks. Upper Saddle River: Pearson, 2004

[142] Nelsen, R. Copulas, characterization, correlation and counter examples. Mathematics Magazine, 1995, Vol. 68, pp. 193-198

[143] Nieddu, L., Manfredi, G., D'Acunto, S. and la null Regina, K. An optimal subclass detection method for credit scoring. World Academy of Science, Engineering and Technology, 2011, Vol. 75, pp. 349-354

[144] Nwulu, N. and Oroja, S. A comparison of different soft computing models for credit scoring. World Academy of Science, Engineering and Technology, 2011, Vol. 78, pp. 898-903

[145] Nwulu, N., Oroja, S. and Ilkan, M. A comparative analysis of machine learning techniques for credit scoring. Information, 2012, Vol. 15(10), pp. 4129-4145

[146] Ong, C.-S., Huang, J.-J. and Tzeng, G.-H. Building credit scoring models using genetic programming. Expert Systems with Applications, 2005, Vol. 29(1), pp. 41-47

[147] Opitz, D. and Maclin, R. Popular ensemble methods: an empirical study. Journal of Artificial Intelligence Research, 1999, Vol. 11, pp. 169-198

[148] Paleologo, G., Elisseeff, A. and Antonini, G. Subagging for credit scoring models. European Journal of Operational Research, 2010, Vol. 201(2), pp. 490-499

[149] Pang, S.-L. Study on credit scoring model and forecasting based on probabilistic neural network. Xitong Gongcheng Lilun yu Shijian/System Engineering Theory and Practice, 2005, Vol. 25(5), pp. 43-48

[150] Pavlidis, N., Tasoulis, D., Adams, N. and Hand, D. Adaptive consumer credit classification. Journal of the Operational Research Society, 2012, Vol. 63(12), pp. 1645-1654

[151] Pearl, J. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Mateo, CA, 1988

[152] Perez, A., Larranaga, P. and Inza, I. Supervised classification with conditional Gaussian networks: increasing the structure complexity from naive Bayes. International Journal of Approximate Reasoning, 2006, Vol. 43(1), pp. 1-25

[153] Ping, Y. and Yongheng, L. Neighborhood rough set and SVM based hybrid credit scoring classifier. Expert Systems with Applications, 2011, Vol. 38(9), pp. 11300-11304

[154] RBNZ, S. Statement of Principles: Bank Registration and Supervision Financial Stability. Banking System Handbook. Wellington: The Reserve Bank of New Zealand, 2013

[155] Rezac, M. Advanced empirical estimate of information value for credit scoring models. Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis, 2011, Vol. 59(2), pp. 267-274

[156] Ripley, B. D. Pattern Recognition and Neural Networks. Cambridge University Press, 1996

[157] Rohit, V. M., Kumar, S. and Kumar, J. Basel II to Basel III ? The Way forward Infosys, 2013

[158] Ruggieri, S., Pedreschi, D. and Turini, F. Data mining for discrimination discovery. ACM Transactions on Knowledge Discovery from Data, 2010, Vol. 4(2), pp. 0.375694444444444-0.402777777777778

[159] Sahami, M. Learning Limited Dependence Bayesian Classifiers. In KDD-96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining AAAI Press, 1996, pp. 335-338

[160] Satterthwaite, S. State Charter School Adoptions: A Probit Regression Model. Politics &; Policy, 2002, Vol. 30, pp. 32-39

[161] Schapire, R. E. The strength of weak learnability. Machine learning, Springer, 1990, Vol. 5(2), pp. 197-227

[162] Seow, H.-V. and Thomas L.C. Using adaptive learning in credit scoring to estimate take-up probability distribution. European Journal of Operational Research, 2006, Vol. 173(3), pp. 880-892

[163] Setiono, R., Baesens, B. and Mues, C. Rule extraction from minimal neural networks for credit card screening. International Journal of Neural Systems, 2011, Vol. 21(4), pp. 265-276

[164] Setiono, R., Baesens, B. and Mues, C. Recursive neural network rule extraction for data with mixed attributes. IEEE Transactions on Neural Networks, 2008, Vol. 19(2), pp. 299-307

[165] Sharma, S. and Osei-Bryson, K.-M. Framework for formal implementation of the business understanding phase of data mining projects. Expert Systems with Applications, 2009, Vol. 36(2 PART 2), pp. 4114-4124

[166] Shen, C., Liu, G. and Deng, N. Improved support vector classification and its application. Jisuanji Gongcheng/Computer Engineering, 2005, Vol. 31(8), pp. 153-154

[167] Shi, Y. Multiple criteria optimization-based data mining methods and applications: A systematic survey. Knowledge and Information Systems, 2010, Vol. 24(3), pp. 369-391

[168] Shi, Y. Current research trend: Information technology and decision making in 2008. International Journal of Information Technology and Decision Making, 2009, Vol. 8(1), pp. 1-5

[169] Sinha, A. and Zhao, H. Incorporating domain knowledge into data mining classifiers: An application in indirect lending. Decision Support Systems, 2008, Vol. 46(1), pp. 287-299

[170] Somol, P., Baesens, B., Pudil, P. and Vanthienen, J. Filter- versus wrapper-based feature selection for credit scoring. International Journal of Intelligent Systems, 2005, Vol. 20(10), pp. 985-999

[171] Sun, C. and Jiang, M. Construction and application of GA-SVM model for personal credit scoring. Journal of Information and Computational Science, 2008, Vol. 5(2), pp. 569-574

[172] Tang, Z., Peng, H. and Yang, B. Investigation and application of cluster analysis in the financial services industry. Journal of Computational Information Systems, 2009, Vol. 5(4), pp. 1363-1368

[173] Thomas, L. Consumer finance: Challenges for operational research. Journal of the Operational Research Society, 2010, Vol. 61(1), pp. 41-52

[174] Thomas, L. C., Edelman, D. and Crook, J. Credit Scoring and its applications SIAM, 2002

[175] Tong, E., Mues, C. and Thomas, L. Mixture cure models in credit scoring: If and when borrowers default. European Journal of Operational Research, 2012, Vol. 218(1), pp. 132-139

[176] Trevor Hastie, Robert Tibshirani, Jerome Friedman. The Elements of Statistical Learning. Springer, 2013, 2 edição.

[177] Tsai, C.-F. Feature selection in bankruptcy prediction. Knowledge-Based Systems, 2009, Vol. 22(2), pp. 120-127

[178] Tsai, C.-F. Financial decision support using neural networks and support vector machines. Expert Systems, 2008, Vol. 25(4), pp. 380-393

[179] Tsai, C.-F. and Wu, J.-W. Using neural network ensembles for bankruptcy prediction and credit scoring. Expert Systems with Applications, 2008, Vol. 34(4), pp. 2639-2649

[180] Van Eyden, R. Statistical modelling versus neural networks in financial decision making. Neural Network World, 1995, Vol. 5(1), pp. 99-108

[181] Van Gestel, T., Martens, D., Baesens, B., Feremans, D., Huysmans, J. and Vanthienen, J. Forecasting and analyzing insurance companies' ratings. International Journal of Forecasting, 2007, Vol. 23(3), pp. 513-529

[182] Van Gool, J., Verbeke, W., Sercu, P. and Baesens, B. Credit scoring for microfinance: Is it worth it? International Journal of Finance and Economics, 2012, Vol. 17(2), pp. 103-123

[183] Van Laere, E. and Baesens B. The development of a simple and intuitive rating system under Solvency II. Insurance: Mathematics and Economics, 2010, Vol. 46(3), pp. 500-510

[184] Vapnik, V. Statistical learning theory. 1998. Wiley, New York, 1998

[185] Verstraeten, G. and Van Den Poel, D. The impact of sample bias on consumer credit scoring performance and profitability. Journal of the Operational Research Society, 2005, Vol. 56(8), pp. 981-992

[186] Vukovic, S., Delibasic, B., Uzelac, A. and Suknovic, M. A case-based reasoning model that uses preference theory functions for credit scoring. Expert Systems with Applications, 2012, Vol. 39(9), pp. 8389-8395

[187] Wang, G., Hao, J., Ma, J. and Jiang, H. A comparative assessment of ensemble learning for credit scoring. Expert Systems with Applications, 2011, Vol. 38(1), pp. 223-230

[188] Wang, G., Ma, J., Huang, L. and Xu, K. Two credit scoring models based on dual strategy ensemble trees. Knowledge-Based Systems, 2012, Vol. 26, pp. 61-68

[189] Wang, J., Hedar, A.-R., Wang, S. and Ma, J. Rough set and scatter search metaheuristic based feature selection for credit scoring. Expert Systems with Applications, 2012, Vol. 39(6), pp. 6123-6128

[190] Wang, Y., Wang, S. and Lai, K. A new fuzzy support vector machine to evaluate credit risk. IEEE Transactions on Fuzzy Systems, 2005, Vol. 13(6), pp. 820-831

[191] Webb, G. I. Naive Bayes. Encyclopedia of Machine Learning, 2010, Vol. 15, pp. 713-714

[192] Webb, G. I., Boughton, J. and Wang, Z. Not So Naive Bayes: Aggregating One-Dependence Estimators. Machine Learning, 2005, Vol. 58, pp. 5-24

[193] West, D. Neural network credit scoring models. Computers and Operations Research, 2000, Vol. 27(11-12), pp. 1131-1152

[194] Witten, I. H., Frank, E. and Hall, M. A. Data Mining: Practical machine learning tools and techniques. 3rd Edition (ed.) Morgan Kaufman, 2011

[195] Wolpert, D. H. Stacked generalization. Neural networks, Elsevier, 1992, Vol. 5(2), pp. 241-259

[196] Won, C., Kim, J. and Bae, J. Using genetic algorithm based knowledge refinement model for dividend policy forecasting. Expert Systems with Applications, 2012, Vol. 39(18), pp. 13472-13479

[197] Wu, W.-W. Improving classification accuracy and causal knowledge for better credit decisions. International Journal of Neural Systems, 2011, Vol. 21(4), pp. 297-309

[198] Xiao, J., Xie, L., He, C. and Jiang, X. Dynamic classifier ensemble model for customer classification with imbalanced class distribution. Expert Systems with Applications, 2012, Vol. 39(3), pp. 3668-3675

[199] Xiao, W., Zhao, Q. and Fei, Q. A comparative study of data mining methods in consumer loans credit scoring management. Journal of Systems Science and Systems Engineering, 2006, Vol. 15(4), pp. 419-435

[200] Xiao, W.-B. and Fei, Q. A study of personal credit scoring models on support vector machine with optimal choice of kernel function parameters. Xitong Gongcheng Lilun yu Shijian/System Engineering Theory and Practice, 2006, Vol. 26(10), pp. 73-79

[201] Xu, X., Zhou, C. and Wang, Z. Credit scoring algorithm based on link analysis ranking with support vector machine. Expert Systems with Applications, 2009, Vol. 36(2 PART 2), pp. 2625-2632

[202] Yang, Y. Adaptive credit scoring with kernel learning methods. European Journal of Operational Research, 2007, Vol. 183(3), pp. 1521-1536

[203] Yang, Z., Wang, Y., Bai, Y. and Zhang, X. Measuring scorecard performance. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2004, Vol. 3039, pp. 900-906

[204] Yap, B., Ong, S. and Husain, N. Using data mining to improve assessment of credit worthiness via credit scoring models. Expert Systems with Applications, 2011, Vol. 38(10), pp. 13274-13283

[205] Yu, J.-L. and Li, H. On performance of feature normalization in classification with distance-based case-based reasoning. Recent Patents on Computer Science, 2011, Vol. 4(3), pp. 203-210

[206] Zadeh, L. A. Fuzzy sets Information and control, Elsevier, 1965, Vol. 8(3), pp. 338-353

[207] Zhang, D., Zhou, X., Leung, S. and Zheng, J. Vertical bagging decision trees model for credit scoring. Expert Systems with Applications, 2010, Vol. 37(12), pp. 7838-7843

[208] Zhang, H.-X. and Mao, Z.-Z. Credit evaluation model based on time series fuzzy clustering and rule extraction. Dongbei Daxue Xuebao/Journal of Northeastern University, 2010, Vol. 31(4), pp. 465-468

[209] Zhou, L., Lai, K. and Yen, J. Credit scoring models with auc maximization based on weighted SVM. International Journal of Information Technology and Decision Making, 2009, Vol. 8(4), pp. 677-696

[210] Zhou, L., Lai, K. and Yu, L. Least squares support vector machines ensemble models for credit scoring. Expert Systems with Applications, 2010, Vol. 37(1), pp. 127-133

[211] Zhu, H., Beling, P. and Overstreet, G. A Bayesian framework for the combination of classifier outputs. Journal of the Operational Research Society, 2002, Vol. 53(7), pp. 719-727

[212] Ziari, H., Leatham, D. and Ellinger, P. Development of statistical discriminant mathematical programming model via resampling estimation techniques. American Journal of Agricultural Economics, 1997, Vol. 79(4), pp. 1352-1362

[213] Zweig, M. H. and Campbell, G. Receiver-operating characteristic (ROC) plots: a fundamental evaluation tool in clinical medicine. Clinical Chemistry, 1993, Vol. 29, pp. 561-577

Apêndice 1 - Lista de artigos utilizados na revisão.

Segue abaixo a lista de artigos utilizados na revisão.

1. Abdou, H. Genetic programming for credit scoring: The case of Egyptian public sector banks. Expert Systems with Applications, 2009, Vol. 36(9), pp. 11402-11417

2. Abdou, H., Pointon, J. and El-Masry, A. Neural nets versus conventional techniques in credit scoring in Egyptian banking. Expert Systems with Applications, 2008, Vol. 35(3), pp. 1275-1292

3. Adams, N., Hand, D. and Till, R. Mining for classes and patterns in behavioural data. Journal of the Operational Research Society, 2001, Vol. 52(9), pp. 1017-1024

4. Akkoc, S. An empirical comparison of conventional techniques, neural networks and the three stage hybrid Adaptive Neuro Fuzzy Inference System (ANFIS) model for credit scoring analysis: The case of Turkish credit card data. European Journal of Operational Research, 2012, Vol. 222(1), pp. 168-178

5. Antonakis, A. and Sfakianakis, M. Assessing naive Bayes as a method for screening credit applicants. Journal of Applied Statistics, 2009, Vol. 36(5), pp. 537-545

6. Baesens, B., Van Gestel, T., Stepanova, M., Van Den Poel, D. and Vanthienen, J. Neural network survival analysis for personal loan data. Journal of the Operational Research Society, 2005, Vol. 56(9), pp. 1089-1098

7. Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J. and Vanthienen, J. Benchmarking state-of-the-art classification algorithms for credit scoring. Journal of the Operational Research Society, 2003, Vol. 54(6), pp. 627-635

8. Banasik, J., Crook, J. and Thomas, L. Sample selection bias in credit scoring models. Journal of the Operational Research Society, 2003, Vol. 54(8), pp. 822-832

9. Banasik, J., Crook, J. and Thomas, L. Not if but when will borrowers default. Journal of the Operational Research Society, 1999, Vol. 50(12), pp. 1185-1190

10. Bardos, M. Detecting the risk of company failure at the Banque de France. Journal of Banking and Finance, 1998, Vol. 22(10-11), pp. 1405-1419

11. Baxter, R., Gawler, M. and Ang, R. Predictive model of insolvency risk for australian corporations. Conferences in Research and Practice in Information Technology Series, 2007, Vol. 70, pp. 21-28

12. Becker, U. and Fahrmeir, L. Bump Hunting for Risk: A New Data Mining Tool and its Applications. Computational Statistics, 2001, Vol. 16(3), pp. 373-386

13. Ben-David, A. and Frank, E. Accuracy of machine learning models versus hand craft. Expert Systems with Applications, 2009, Vol. 36(3 PART 1), pp. 5264-5271

14. Berger, A., Frame, W. and Miller, N. Credit scoring and the availability, price, and risk of small business credit. Journal of Money, Credit and Banking, 2005, Vol. 37(2), pp. 191-222

15. Bijak, K. and Thomas, L. Does segmentation always improve model performance in credit scoring? Expert Systems with Applications, 2012, Vol. 39(3), pp. 2433-2442

16. Brown, I. and Mues, C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Systems with Applications, 2012, Vol. 39(3), pp. 3446-3453

17. Burton, D. Credit scoring, risk, and consumer lendingscapes in emerging markets. Environment and Planning A, ¡b¿2012¡/b¿, Vol. 44(1), pp. 111-124

18. Capotorti, A. and Barbanera, E. Credit scoring analysis using a fuzzy probabilistic rough set model. Computational Statistics and Data Analysis, 2012, Vol. 56(4), pp. 981-994

19. Chang, S.-Y. and Yeh, T.-Y. An artificial immune classifier for credit scoring analysis. Applied Soft Computing Journal, 2012, Vol. 12(2), pp. 611-618

20. Chen, F.-L. and Li, F.-C. Combination of feature selection approaches with SVM in credit scoring. Expert Systems with Applications, 2010, Vol. 37(7), pp. 4902-4909

21. Chen, M.-C. and Huang, S.-H. Credit scoring and rejected instances reassigning through evolutionary computation techniques. Expert Systems with Applications, 2003, Vol. 24(4), pp. 433-441

22. Chen, W., Ma, C. and Ma, L. Mining the customer credit using hybrid support vector machine technique. Expert Systems with Applications, 2009, Vol. 36(4), pp. 7611-7616

23. Chrzanowska, M., Alfaro, E. and Witkowska, D. The individual borrowers recognition: Single and ensemble trees. Expert Systems with Applications, 2009, Vol. 36(3 PART 2), pp. 6409-6414

24. Chuang, C.-L. and Huang, S.-T. A hybrid neural network approach for credit scoring. Expert Systems, 2011, Vol. 28(2), pp. 185-196

25. Chuang, C.-L. and Lin, R.-H. Constructing a reassigning credit scoring model. Expert Systems with Applications, 2009, Vol. 36(2 PART 1), pp. 1685-1694

26. DeYoung, R., Frame, W., Glennon, D. and Nigro, P. The Information Revolution and Small Business Lending: The Missing Evidence. Journal of Financial Services Research, 2011, Vol. 39(41306), pp. 19-33

27. Dong, Y., Hao, X. and Yu, C. Comparison of statistical and artificial intelligence methodologies in small-businesses' credit assessment based on daily transaction data. ICIC Express Letters, 2011, Vol. 5(5), pp. 1725-1730

28. Dryver, A. and Sukkasem, J. Validating risk models with a focus on credit scoring models. Journal of Statistical Computation and Simulation, 2009, Vol. 79(2), pp. 181-193

29. Efromovich, S. Oracle inequality for conditional density estimation and an actuarial example. Annals of the Institute of Statistical Mathematics, 2010, Vol. 62(2), pp. 249-275

30. Ewout W. Steyerberg Andrew J. Vickers, N. R. C. T. G. M. G. N. O. M. J. P. and Kattan, M. W. Assessing the performance of prediction models: a framework for some traditional and novel measures. NIH Public Access, 2010, Vol. 21(1), pp. 128-138

31. Falangis, K. and Glen, J. Heuristics for feature selection in mathematical programming discriminant analysis models. Journal of the Operational Research Society, 2010, Vol. 61(5), pp. 804-812

32. Feng, L., Yao, Y. and Jin, B. Research on credit scoring model with SVM for network management. Journal of Computational Information Systems, 2010, Vol. 6(11), pp. 3567-3574

33. Figini, S. and Uberti, P. Model assessment for predictive classification models. Communications in Statistics - Theory and Methods, 2010, Vol. 39(18), pp. 3238-3244

34. Finlay, S. Multiple classifier architectures and their application to credit risk assessment. European Journal of Operational Research, 2011, Vol. 210(2), pp. 368-378

35. Finlay, S. Credit scoring for profitability objectives. European Journal of Operational Research, 2010, Vol. 202(2), pp. 528-537

36. Finlay, S. Are we modelling the right thing? The impact of incorrect problem specification in credit scoring. Expert Systems with Applications, 2009, Vol. 36(5), pp. 9065-9071

37. Finlay, S. Towards profitability: A utility approach to the credit scoring problem. Journal of the Operational Research Society, 2008, Vol. 59(7), pp. 921-931

38. Gemela, J. Financial analysis using Bayesian networks. Applied Stochastic Models in Business and Industry, 2001, Vol. 17(1), pp. 57-67

39. Gestel, T., Baesens, B., Suykens, J., Van den Poel, D., Baestaens, D.-E. and Willekens, M. Bayesian kernel based classification for financial distress detection. European Journal of Operational Research, 2006, Vol. 172(3), pp. 979-1003

40. Giudici, P. Bayesian data mining, with application to benchmarking and credit scoring. Applied Stochastic Models in Business and Industry, 2001, Vol. 17(1), pp. 69-81

41. Hand, D. Supervised classification and tunnel vision. Applied Stochastic Models in Business and Industry, 2005, Vol. 21(2), pp. 97-109

42. Hand, D. Good practice in retail credit scorecard assessment. Journal of the Operational Research Society, 2005, Vol. 56(9), pp. 1109-1117

43. Hand, D. Modelling consumer credit risk. IMA Journal Management Mathematics, 2001, Vol. 12(2), pp. 139-155

44. ¡Hand, D. and Henley, W. Statistical classification methods in consumer credit scoring: A review. Journal of the Royal Statistical Society. Series A: Statistics in Society, 1997, Vol. 160(3), pp. 523-541

45. Hand, D. and Kelly, M. Superscorecards. IMA Journal Management Mathematics, 2002, Vol. 13(4), pp. 273-281

46. Hardle, W., Mammen, E. and Muller, M. Testing parametric versus semiparametric modeling in generalized linear models. Journal of the American Statistical Association, 1998, Vol. 93(444), pp. 1461-1474

47. He, J., Zhang, Y., Shi, Y. and Huang, G. Domain-driven classification based on multiple criteria and multiple constraint-level programming for intelligent credit scoring. IEEE Transactions on Knowledge and Data Engineering, 2010, Vol. 22(6), pp. 826-838

48. Hens, A. and Tiwari, M. Computational time reduction for credit scoring: An integrated approach based on support vector machine and stratified sampling method. Expert Systems with Applications, 2012, Vol. 39(8), pp. 6774-6781

49. Hoffmann, F., Baesens, B., Martens, J., Put, F. and Vanthienen, J. Comparing a genetic fuzzy and a neurofuzzy classifier for credit scoring. International Journal of Intelligent Systems, 2002, Vol. 17(11), pp. 1067-1083

50. Hoffmann, F., Baesens, B., Mues, C., Van Gestel, T. and Vanthienen, J. Inferring descriptive and approximate fuzzy rules for credit scoring using evolutionary algorithms. European Journal of Operational Research, 2007, Vol. 177(1), pp. 540-555

51. Hsieh, N.-C. Hybrid mining approach in the design of credit scoring models. Expert Systems with Applications, 2005, Vol. 28(4), pp. 655-665

52. Hsieh, N.-C. and Hung, L.-P. A data driven ensemble classifier for credit scoring analysis. Expert Systems with Applications, 2010, Vol. 37(1), pp. 534-545

53. Hu, Y.-C. and Ansell, J. Retail default prediction by using sequential minimal optimization technique. Journal of Forecasting, 2009, Vol. 28(8), pp. 651-666

54. Hu, Y.-C. and Ansell, J. Measuring retail company performance using credit scoring techniques. European Journal of Operational Research, 2007, Vol. 183(3), pp. 1595-1606

55. Huang, C.-L., Chen, M.-C. and Wang, C.-J. Credit scoring with a data mining approach based on support vector machines. Expert Systems with Applications, 2007, Vol. 33(4), pp. 847-856

56. Huang, J.-J., Tzeng, G.-H. and Ong, C.-S. Two-stage genetic programming (2SGP) for the credit scoring model. Applied Mathematics and Computation, 2006, Vol. 174(2), pp. 1039-1053

57. Huang, Y.-M., Hung, C.-M. and Jiau, H. Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem. Nonlinear Analysis: Real World Applications, 2006, Vol. 7(4), pp. 720-747

58. Huysmans, J., Baesens, B., Vanthienen, J. and Van Gestel, T. Failure prediction with self organizing maps. Expert Systems with Applications, 2006, Vol. 30(3), pp. 479-487

59. Ince, H. and Aktan, B. A comparison of data mining techniques for credit scoring in banking: A managerial perspective. Journal of Business Economics and Management, 2009, Vol. 10(3), pp. 233-240

60. John, G., Miller, P. and Kerber, R. Stock selection using rule induction. IEEE Expert-Intelligent Systems and their Applications, 1996, Vol. 11(5), pp. 52-58

61. Jung, K. and Thomas, L. A note on coarse classifying in acceptance scorecards. Journal of the Operational Research Society, 2008, Vol. 59(5), pp. 714-718

62. Kao, L.-J., Chiu, C.-C. and Chiu, F.-Y. A Bayesian latent variable model with classification and regression tree approach for behavior and credit scoring. Knowledge-Based Systems, 2012, Vol. 36, pp. 245-252

63. Karlis, D. and Rahmouni, M. Analysis of defaulters' behaviour using the Poisson-mixture approach. IMA Journal Management Mathematics, 2007, Vol. 18(3), pp. 297-311

64. Kocenda, E. and Vojtek, M. Default predictors in retail credit scoring: Evidence from Czech banking data. Emerging Markets Finance and Trade, 2011, Vol. 47(6), pp. 80-98

65. Laha, A. Building contextual classifiers by integrating fuzzy rule based classification technique and k-nn method for credit scoring. Advanced Engineering Informatics, 2007, Vol. 21(3), pp. 281-291

66. Lahsasna, A., Ainon, R. and Wah, T. Enhancement of transparency and accuracy of credit scoring models through genetic fuzzy classifier. Maejo International Journal of Science and Technology, 2010, Vol. 4(1), pp. 136-158

67. Lahsasna, A., Ainon, R. and Wah, T. Credit scoring models using soft computing methods: A survey. International Arab Journal of Information Technology, 2010, Vol. 7(2), pp. 115-123

68. Lan, Y., Janssens, D., Chen, G. and Wets, G. Improving associative classification by incorporating novel interestingness measures. Expert Systems with Applications, 2006, Vol. 31(1), pp. 184-192

69. Lee, T.-S. and Chen, I.-F. A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. Expert Systems with Applications, 2005, Vol. 28(4), pp. 743-752

70. Lee, T.-S., Chiu, C.-C., Chou, Y.-C. and Lu, C.-J. Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. Computational Statistics and Data Analysis, 2006, Vol. 50(4), pp. 1113-1130

71. Lee, T.-S., Chiu, C.-C., Lu, C.-J. and Chen, I.-F. Credit scoring using the hybrid neural discriminant technique. Expert Systems with Applications, 2002, Vol. 23(3), pp. 245-254

72. Li, H. and Hand, D. Direct versus indirect credit scoring classifications. Journal of the Operational Research Society, 2002, Vol. 53(6), pp. 647-654

73. Li, S.-T., Shiue, W. and Huang, M.-H. The evaluation of consumer loans using support vector machines. Expert Systems with Applications, 2006, Vol. 30(4), pp. 772-782

74. Ling, Y., Cao, Q. and Zhang, H. Credit scoring using multi-kernel support vector machine and chaos particle swarm optimization. International Journal of Computational Intelligence and Applications, 2012, Vol. 11(3), pp. 12500198:1-12500198:13

75. Lisboa, P., Etchells, T., Jarman, I., Arsene, C., Aung, M., Eleuteri, A., Taktak, A., Ambrogi, F., Boracchi, P. and Biganzoli, E. Partial logistic artificial neural network for competing risks regularized with automatic relevance determination. IEEE Transactions on Neural Networks, 2009, Vol. 20(9), pp. 1403-1416

76. Liu, X., Fu, H. and Lin, W. A modified support vector machine model for credit scoring. International Journal of Computational Intelligence Systems, 2010, Vol. 3(6), pp. 797-803

77. Liu, Y. and Schumann, M. Data mining feature selection for credit scoring models. Journal of the Operational Research Society, 2005, Vol. 56(9), pp. 1099-1108

78. Louzada, F., Anacleto-Junior, O., Candolo, C. and Mazucheli, J. Poly-bagging predictors for classification modelling for credit scoring. Expert Systems with Applications, 2011, Vol. 38(10), pp. 12717-12720

79. Louzada, F., Cancho, V., Roman, M. and Leite, J. A new long-term lifetime distribution induced by a latent complementary risk framework. Journal of Applied Statistics, 2012, Vol. 39(10), pp. 2209-2222

80. Louzada, F., Ferreira-Silva, P. and Diniz, C. On the impact of disproportional samples in credit scoring models: An application to a Brazilian bank data. Expert Systems with Applications, 2012, Vol. 39(9), pp. 8071-8078

81. Lucas, A. Statistical challenges in credit card issuing. Applied Stochastic Models in Business and Industry, 2001, Vol. 17(1), pp. 69-81

82. Luo, S.-T., Cheng, B.-W. and Hsieh, C.-H. Prediction model building with clustering-launched classification and support vector machines in credit scoring. Expert Systems with Applications, 2009, Vol. 36(4), pp. 7562-7566

83. Marcano-Cedeno, A., Marin-De-La-Barcena, A., Jimenez-Trillo, J., Pinuela, J. and Andina, D. Artificial metaplasticity neural network applied to credit scoring. International Journal of Neural Systems, 2011, Vol. 21(4), pp. 311-317

84. Marques, A., Garcia, V. and Sanchez, J. Exploring the behaviour of base classifiers in credit scoring ensembles. Expert Systems with Applications, 2012, Vol. 39(11), pp. 10244-10250

85. Marques, A., Garcia, V. and Sanchez, J. Two-level classifier ensembles for credit risk assessment. Expert Systems with Applications, 2012, Vol. 39(12), pp. 10916-10922

86. Marron, D. 'Lending by numbers': Credit scoring and the constitution of risk within American consumer credit. Economy and Society, 2007, Vol. 36(1), pp. 103-133

87. Martens, D., Baesens, B., Van Gestel, T. and Vanthienen, J. Comprehensible credit scoring models using rule extraction from support vector machines. European Journal of Operational Research, 2007, Vol. 183(3), pp. 1466-1476

88. Martens, D., Van Gestel, T., De Backer, M., Haesen, R., Vanthienen, J. and Baesens, B. Credit rating prediction using Ant Colony Optimization. Journal of the Operational Research Society, 2010, Vol. 61(4), pp. 561-573

89. Maznevski, M., Kemp, R., Overstreet, G. and Crook, J. The power to borrow and lend: investigating the cultural context as part of the lending decision. Journal of the Operational Research Society, 2001, Vol. 52(9), pp. 1045-1056

90. McDonald, R., Sturgess, M., Smith, K., Hawkins, M. and Huang, E. Non-linearity of scorecard log-odds. International Journal of Forecasting, 2012, Vol. 28(1), pp. 239-247

91. Mues, C., Baesens, B., Files, C. and Vanthienen, J. Decision diagrams in machine learning: An empirical study on real-life credit-risk data. Expert Systems with Applications, 2004, Vol. 27(2), pp. 257-264

92. Nieddu, L., Manfredi, G., D'Acunto, S. and la null Regina, K. An optimal subclass detection method for credit scoring. World Academy of Science, Engineering and Technology, 2011, Vol. 75, pp. 349-354

93. Nwulu, N. and Oroja, S. A comparison of different soft computing models for credit scoring. World Academy of Science, Engineering and Technology, 2011, Vol. 78, pp. 898-903

94. Ong, C.-S., Huang, J.-J. and Tzeng, G.-H. Building credit scoring models using genetic programming. Expert Systems with Applications, 2005, Vol. 29(1), pp. 41-47

95. Paleologo, G., Elisseeff, A. and Antonini, G. Subagging for credit scoring models. European Journal of Operational Research, 2010, Vol. 201(2), pp. 490-499

96. Pavlidis, N., Tasoulis, D., Adams, N. and Hand, D. Adaptive consumer credit classification. Journal of the Operational Research Society, 2012, Vol. 63(12), pp. 1645-1654

97. Ping, Y. and Yongheng, L. Neighborhood rough set and SVM based hybrid credit scoring classifier. Expert Systems with Applications, 2011, Vol. 38(9), pp. 11300-11304

98. Rezac, M. Advanced empirical estimate of information value for credit scoring models. Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis, 2011, Vol. 59(2), pp. 267-274

99. Ruggieri, S., Pedreschi, D. and Turini, F. Data mining for discrimination discovery. ACM Transactions on Knowledge Discovery from Data, 2010, Vol. 4(2), pp. 0.375694444444444-0.402777777777778

100. Setiono, R., Baesens, B. and Mues, C. Rule extraction from minimal neural networks for credit card screening. International Journal of Neural Systems, 2011, Vol. 21(4), pp. 265-276

101. Setiono, R., Baesens, B. and Mues, C Recursive neural network rule extraction for data with mixed attributes. IEEE Transactions on Neural Networks, 2008, Vol. 19(2), pp. 299-307

102. Sharma, S. and Osei-Bryson, K.-M. Framework for formal implementation of the business understanding phase of data mining projects. Expert Systems with Applications, 2009, Vol. 36(2 PART 2), pp. 4114-4124

103. Shi, Y. Multiple criteria optimization-based data mining methods and applications: A systematic survey. Knowledge and Information Systems, 2010, Vol. 24(3), pp. 369-391

104. Shi, Y. Current research trend: Information technology and decision making in 2008. International Journal of Information Technology and Decision Making, 2009, Vol. 8(1), pp. 1-5

105. Sinha, A. and Zhao, H. Incorporating domain knowledge into data mining classifiers: An application in indirect lending. Decision Support Systems, 2008, Vol. 46(1), pp. 287-299

106. Somol, P., Baesens, B., Pudil, P. and Vanthienen, J. Filter- versus wrapper-based feature selection for credit scoring. International Journal of Intelligent Systems, 2005, Vol. 20(10), pp. 985-999

107. Thomas, L. Consumer finance: Challenges for operational research. Journal of the Operational Research Society, 2010, Vol. 61(1), pp. 41-52

108. Tong, E., Mues, C. and Thomas, L. Mixture cure models in credit scoring: If and when borrowers default. European Journal of Operational Research, 2012, Vol. 218(1), pp. 132-13

109. Tsai, C.-F. Feature selection in bankruptcy prediction. Knowledge-Based Systems, 2009, Vol. 22(2), pp. 120-127

110. Tsai, C.-F. Financial decision support using neural networks and support vector machines. Expert Systems, 2008, Vol. 25(4), pp. 380-393

111. Tsai, C.-F. and Wu, J.-W. Using neural network ensembles for bankruptcy prediction and credit scoring. Expert Systems with Applications, 2008, Vol. 34(4), pp. 2639-2649

112. Van Gestel, T., Martens, D., Baesens, B., Feremans, D., Huysmans, J. and Vanthienen, J. Forecasting and analyzing insurance companies' ratings. International Journal of Forecasting, 2007, Vol. 23(3), pp. 513-529

113. Van Gool, J., Verbeke, W., Sercu, P. and Baesens, B. Credit scoring for microfinance: Is it worth it? International Journal of Finance and Economics, 2012, Vol. 17(2), pp. 103-123

114. Verstraeten, G. and Van Den Poel, D. The impact of sample bias on consumer credit scoring performance and profitability. Journal of the Operational Research Society, 2005, Vol. 56(8), pp. 981-992

115. Vukovic, S., Delibasic, B., Uzelac, A. and Suknovic, M. A case-based reasoning model that uses preference theory functions for credit scoring. Expert Systems with Applications, 2012, Vol. 39(9), pp. 8389-8395

116. Wang, G., Hao, J., Ma, J. and Jiang, H. A comparative assessment of ensemble learning for credit scoring. Expert Systems with Applications, 2011, Vol. 38(1), pp. 223-230

117. Wang, G., Ma, J., Huang, L. and Xu, K. Two credit scoring models based on dual strategy ensemble trees. Knowledge-Based Systems, 2012, Vol. 26, pp. 61-68

118. Wang, J., Hedar, A.-R., Wang, S. and Ma, J. Rough set and scatter search metaheuristic based feature selection for credit scoring. Expert Systems with Applications, 2012, Vol. 39(6), pp. 6123-6128

119. Wang, Y., Wang, S. and Lai, K. A new fuzzy support vector machine to evaluate credit risk. IEEE Transactions on Fuzzy Systems, 2005, Vol. 13(6), pp. 820-831

120. West, D. Neural network credit scoring models. Computers and Operations Research, 2000, Vol. 27(11-12), pp. 1131-1152

121. Won, C., Kim, J. and Bae, J. Using genetic algorithm based knowledge refinement model for dividend policy forecasting. Expert Systems with Applications, 2012, Vol. 39(18), pp. 13472-13479

122. Wu, W.-W. Improving classification accuracy and causal knowledge for better credit decisions. International Journal of Neural Systems, 2011, Vol. 21(4), pp. 297-309

123. Xiao, J., Xie, L., He, C. and Jiang, X. Dynamic classifier ensemble model for customer classification with imbalanced class distribution. Expert Systems with Applications, 2012, Vol. 39(3), pp. 3668-3675

124. Xiao, W., Zhao, Q. and Fei, Q. A comparative study of data mining methods in consumer loans credit scoring management. Journal of Systems Science and Systems Engineering, 2006, Vol. 15(4), pp. 419-435

125. Xu, X., Zhou, C. and Wang, Z. Credit scoring algorithm based on link analysis ranking with support vector machine. Expert Systems with Applications, 2009, Vol. 36(2 PART 2), pp. 2625-2632

126. Yang, Y. Adaptive credit scoring with kernel learning methods. European Journal of Operational Research, 2007, Vol. 183(3), pp. 1521-1536

127. Yang, Z., Wang, Y., Bai, Y. and Zhang, X. Measuring scorecard performance. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2004, Vol. 3039, pp. 900-906

128. Yap, B., Ong, S. and Husain, N. Using data mining to improve assessment of credit worthiness via credit scoring models. Expert Systems with Applications, 2011, Vol. 38(10), pp. 13274-13283

129. Yu, J.-L. and Li, H. On performance of feature normalization in classification with distance-based case-based reasoning. Recent Patents on Computer Science, 2011, Vol. 4(3), pp. 203-210

130. Zhang, D., Zhou, X., Leung, S. and Zheng, J. Vertical bagging decision trees model for credit scoring. Expert Systems with Applications, 2010, Vol. 37(12), pp. 7838-7843

131. Zhou, L., Lai, K. and Yen, J. Credit scoring models with auc maximization based on weighted SVM. International Journal of Information Technology and Decision Making, 2009, Vol. 8(4), pp. 677-696

132. Zhou, L., Lai, K. and Yu, L. Least squares support vector machines ensemble models for credit scoring. Expert Systems with Applications, 2010, Vol. 37(1), pp. 127-133

133. Zhu, H., Beling, P. and Overstreet, G. A Bayesian framework for the combination of classifier outputs. Journal of the Operational Research Society, 2002, Vol. 53(7), pp. 719-727

134. Ziari, H., Leatham, D. and Ellinger, P. Development of statistical discriminant mathematical programming model via resampling estimation techniques. American Journal of Agricultural Economics, 1997, Vol. 79(4), pp. 1352-1362

Apêndice 2 - Códigos R

Seguem alguns códigos em R utilizados nas simulações. Ao observar os códigos nota-se que são muito parecidos, a ideia é essa, ou seja, padronizar o procedimento.

Código para rede neural.

```
library("MASS")
library("nnet")
library(AUC)

mubom = c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
mumau = c(1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),
1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20))

sigmabom = matrix(0,20,20)
sigmamau = matrix(0,20,20)

sigmabom[1,1]=4
sigmabom[2,2]=4
sigmabom[3,3]=4
sigmabom[4,4]=4
sigmabom[5,5]=4
sigmabom[6,6]=4
sigmabom[7,7]=4
sigmabom[8,8]=4
sigmabom[9,9]=4
sigmabom[10,10]=4
sigmabom[11,11]=4
sigmabom[12,12]=4
sigmabom[13,13]=4
sigmabom[14,14]=4
sigmabom[15,15]=4
sigmabom[16,16]=4
sigmabom[17,17]=4
sigmabom[18,18]=4
sigmabom[19,19]=4
sigmabom[20,20]=4

sigmamau[1,1]=1
sigmamau[2,2]=1
sigmamau[3,3]=1
sigmamau[4,4]=1
sigmamau[5,5]=1
sigmamau[6,6]=1
sigmamau[7,7]=1
sigmamau[8,8]=1
sigmamau[9,9]=1
sigmamau[10,10]=1
sigmamau[11,11]=1
sigmamau[12,12]=1
sigmamau[13,13]=1
sigmamau[14,14]=1
sigmamau[15,15]=1
sigmamau[16,16]=1
sigmamau[17,17]=1
sigmamau[18,18]=1
sigmamau[19,19]=1
sigmamau[20,20]=1

vacc = matrix(0,100,1)
vbB = matrix(0,100,1)
vmM = matrix(0,100,1)
vauc = matrix(0,100,1)
```

```
for (j in 1:100)
{
n=100000

r = matrix(0,n,1)
x=matrix(0,n,20)

propbons=0.9
propmaus=0.1

bons=mvrnorm(propbons*n,mubom, sigmabom)
maus=mvrnorm(propmaus*n,mumau, sigmamau)

aux=propbons*n
for(i in 1:aux)
{
r[i] = 0
x[i,]=bons[i,]
}
aux=propbons*n + 1
for(i in aux:n)
{
r[i] = 1
x[i,]=maus[i-aux+1,]
}


#############AJUSTAR REDE NEURAL##################

r=factor(r)
dados = data.frame(r,x)
modeloneural = nnet(r ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10 + X11 + X12 + X13 + X14 + X15 + X16 + X17 + X18 + X19 + X20, data = dados,
size=9, maxit=1000)

################################################

n = 20000

propbons = 0.5
propmaus = 0.5

bons2=mvrnorm(propbons*n,mubom, sigmabom)
maus2=mvrnorm(propmaus*n,mumau, sigmamau)

r = matrix(0,n,1)
x = matrix(0,n,20)

aux=propbons*n
for(i in 1:aux)
{
r[i] = 0
x[i,]=bons2[i,]
}
aux=propbons*n + 1
for(i in aux:n)
{
r[i] = 1
x[i,]=maus2[i-aux+1,]
}

dados = data.frame(x)

Dpdct=predict(modeloneural, dados, type="class")

acc=0
mM=0
bB=0
M=0 ##Quantidade de inadimplentes
```

```
B=0 ##Quantidade de adimplentes

for(i in 1:n)
{
if (r[i]==Dpdct[i])
{
acc = acc + 1
}
if (r[i]==1 && Dpdct[i]==1)
{
mM = mM + 1
}
if (r[i]==0 && Dpdct[i]==0)
{
bB = bB + 1
}
if (r[i]==1)
{
M = M + 1
}
if (r[i]==0)
{
B = B + 1
}
}
vacc[j]=acc/n
vmM[j] = mM/M
vbB[j]=bB/B

Cpdct=predict(modeloneural, dados)
r=factor(r)
vauc[j]=auc(roc(Cpdct,r))

}

write("Simulacao maus- bons(10-90 - rede neural)","C:/Users/Renato/Desktop/codigos dissertacao/dados_9nn_10_90.txt", append = TRUE)

write(" ","C:/Users/Renato/Desktop/codigos dissertacao/dados_9nn_10_90.txt", append = TRUE)
write(" ","C:/Users/Renato/Desktop/codigos dissertacao/dados_9nn_10_90.txt", append = TRUE)

write("##############--ACC--##################","C:/Users/Renato/Desktop/codigos dissertacao/dados_9nn_10_90.txt", append = TRUE)

write(vacc,"C:/Users/Renato/Desktop/codigos dissertacao/dados_9nn_10_90.txt", append = TRUE)

write("##############--bB--##################","C:/Users/Renato/Desktop/codigos dissertacao/dados_9nn_10_90.txt", append = TRUE)

write(vbB,"C:/Users/Renato/Desktop/codigos dissertacao/dados_9nn_10_90.txt", append = TRUE)

write("##############--mM--##################","C:/Users/Renato/Desktop/codigos dissertacao/dados_9nn_10_90.txt", append = TRUE)

write(vmM,"C:/Users/Renato/Desktop/codigos dissertacao/dados_9nn_10_90.txt", append = TRUE)

write("##############--auc--##################","C:/Users/Renato/Desktop/codigos dissertacao/dados_9nn_10_90.txt", append = TRUE)

write(vauc,"C:/Users/Renato/Desktop/codigos dissertacao/dados_9nn_10_90.txt", append = TRUE)
```

Código para árvore.

```
library("MASS")
library("rpart")
library(AUC)

n=100000

mubom = c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
mumau = c(1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),
1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20))

sigmabom = matrix(0,20,20)
sigmamau = matrix(0,20,20)

sigmabom[1,1]=4
sigmabom[2,2]=4
sigmabom[3,3]=4
sigmabom[4,4]=4
sigmabom[5,5]=4
sigmabom[6,6]=4
sigmabom[7,7]=4
sigmabom[8,8]=4
sigmabom[9,9]=4
sigmabom[10,10]=4
sigmabom[11,11]=4
sigmabom[12,12]=4
sigmabom[13,13]=4
sigmabom[14,14]=4
sigmabom[15,15]=4
sigmabom[16,16]=4
sigmabom[17,17]=4
sigmabom[18,18]=4
sigmabom[19,19]=4
sigmabom[20,20]=4

sigmamau[1,1]=1
sigmamau[2,2]=1
sigmamau[3,3]=1
sigmamau[4,4]=1
sigmamau[5,5]=1
sigmamau[6,6]=1
sigmamau[7,7]=1
sigmamau[8,8]=1
sigmamau[9,9]=1
sigmamau[10,10]=1
sigmamau[11,11]=1
sigmamau[12,12]=1
sigmamau[13,13]=1
sigmamau[14,14]=1
sigmamau[15,15]=1
sigmamau[16,16]=1
sigmamau[17,17]=1
sigmamau[18,18]=1
sigmamau[19,19]=1
sigmamau[20,20]=1

vacc = matrix(0,100,1)
vbB = matrix(0,100,1)
vmM = matrix(0,100,1)
vauc = matrix(0,100,1)

for(j in 1:100)
{

n=100000

r = matrix(0,n,1)
```

```
x=matrix(0,n,20)

propbons=0.9
propmaus=0.1

bons=mvrnorm(propbons*n,mubom, sigmabom)
maus=mvrnorm(propmaus*n,mumau, sigmamau)

aux=propbons*n
for(i in 1:aux)
{
r[i] = 0
x[i,]=bons[i,]
}
aux=propbons*n + 1
for(i in aux:n)
{
r[i] = 1
x[i,]=maus[i-aux+1,]
}

dados = data.frame(r,x)

Dtree = rpart(r ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10 + X11 + X12 + X13 + X14 + X15 + X16 + X17 + X18 + X19 + X20, data = dados, method = "class")
Ctree = rpart(r ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10 + X11 + X12 + X13 + X14 + X15 + X16 + X17 + X18 + X19 + X20, data = dados)

Dptree = prune(Dtree, cp=Dtree$cptable[which.min(Dtree$cptable[,"xerror"]),"CP"])
Cptree = prune(Ctree, cp=Ctree$cptable[which.min(Ctree$cptable[,"xerror"]),"CP"])

##########################--Teste de modelo--############################################

n = 20000

propbons = 0.5
propmaus = 0.5

bons2=mvrnorm(propbons*n,mubom, sigmabom)
maus2=mvrnorm(propmaus*n,mumau, sigmamau)

r = matrix(0,n,1)
x = matrix(0,n,20)

aux=propbons*n
for(i in 1:aux)
{
r[i] = 0
x[i,]=bons2[i,]
}
aux=propbons*n + 1
for(i in aux:n)
{
r[i] = 1
x[i,]=maus2[i-aux+1,]
}

dados = data.frame(x)


Dpdct = predict(Dptree, newdata=dados, type="class")

acc=0
mM=0
bB=0
M=0 ##Quantidade de inadimplentes
B=0 ##Quantidade de adimplentes

for(i in 1:n)
{
```

```
if (r[i]==Dpdct[i])
{
acc = acc + 1
}
if (r[i]==1 && Dpdct[i]==1)
{
mM = mM + 1
}
if (r[i]==0 && Dpdct[i]==0)
{
bB = bB + 1
}
if (r[i]==1)
{
M = M + 1
}
if (r[i]==0)
{
B = B + 1
}
}
vacc[j]=acc/n
vmM[j] = mM/M
vbB[j]=bB/B

Cpdct = predict(Cptree, newdata=dados)
r=factor(r)

vauc[j]=auc(roc(Cpdct,r))

}

write("Simulacao maus- bons(40-60 - árvore)","C:/Users/Renato/Desktop/codigos dissertacao/dados_arvore_10_90.txt", append = TRUE)

write(" ","C:/Users/Renato/Desktop/codigos dissertacao/dados_arvore_10_90.txt", append = TRUE)
write(" ","C:/Users/Renato/Desktop/codigos dissertacao/dados_arvore_10_90.txt", append = TRUE)

write("#############--ACC--#################","C:/Users/Renato/Desktop/codigos dissertacao/dados_arvore_10_90.txt", append = TRUE)

write(vacc,"C:/Users/Renato/Desktop/codigos dissertacao/dados_arvore_10_90.txt", append = TRUE)

write("#############--bB--#################","C:/Users/Renato/Desktop/codigos dissertacao/dados_arvore_10_90.txt", append = TRUE)

write(vbB,"C:/Users/Renato/Desktop/codigos dissertacao/dados_arvore_10_90.txt", append = TRUE)

write("#############--mM--#################","C:/Users/Renato/Desktop/codigos dissertacao/dados_arvore_10_90.txt", append = TRUE)

write(vmM,"C:/Users/Renato/Desktop/codigos dissertacao/dados_arvore_10_90.txt", append = TRUE)

write("#############--auc--#################","C:/Users/Renato/Desktop/codigos dissertacao/dados_arvore_10_90.txt", append = TRUE)

write(vauc,"C:/Users/Renato/Desktop/codigos dissertacao/dados_arvore_10_90.txt", append = TRUE)
```

Código para regressão logística.

```
library("MASS")
library(AUC)

mubom = c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
mumau = c(1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),
1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20))
sigmabom = matrix(0,20,20)
sigmamau = matrix(0,20,20)

sigmabom[1,1]=4
sigmabom[2,2]=4
sigmabom[3,3]=4
sigmabom[4,4]=4
sigmabom[5,5]=4
sigmabom[6,6]=4
sigmabom[7,7]=4
sigmabom[8,8]=4
sigmabom[9,9]=4
sigmabom[10,10]=4
sigmabom[11,11]=4
sigmabom[12,12]=4
sigmabom[13,13]=4
sigmabom[14,14]=4
sigmabom[15,15]=4
sigmabom[16,16]=4
sigmabom[17,17]=4
sigmabom[18,18]=4
sigmabom[19,19]=4
sigmabom[20,20]=4

sigmamau[1,1]=1
sigmamau[2,2]=1
sigmamau[3,3]=1
sigmamau[4,4]=1
sigmamau[5,5]=1
sigmamau[6,6]=1
sigmamau[7,7]=1
sigmamau[8,8]=1
sigmamau[9,9]=1
sigmamau[10,10]=1
sigmamau[11,11]=1
sigmamau[12,12]=1
sigmamau[13,13]=1
sigmamau[14,14]=1
sigmamau[15,15]=1
sigmamau[16,16]=1
sigmamau[17,17]=1
sigmamau[18,18]=1
sigmamau[19,19]=1
sigmamau[20,20]=1

vacc = matrix(0,100,1)
vbB = matrix(0,100,1)
vmM = matrix(0,100,1)
vauc = matrix(0,100,1)

for (j in 1:100)
{

n=100000
r = matrix(0,n,1)
x=matrix(0,n,20)

propbons=0.9
propmaus=0.1
```

```
bons=mvrnorm(propbons*n,mubom, sigmabom)
maus=mvrnorm(propmaus*n,mumau, sigmamau)

aux=propbons*n
for(i in 1:aux)
{
r[i] = 0
x[i,]=bons[i,]
}
aux=propbons*n + 1
for(i in aux:n)
{
r[i] = 1
x[i,]=maus[i-aux+1,]
}

dados = data.frame(r,x)

modelo=glm(r ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10 + X11 + X12 + X13 + X14 + X15 + X16 + X17 + X18 + X19 + X20, data = dados ,
family=binomial(link = "logit"))

##########################--Teste de modelo--############################################

n = 20000

propbons = 0.5
propmaus = 0.5

bons2=mvrnorm(propbons*n,mubom, sigmabom)
maus2=mvrnorm(propmaus*n,mumau, sigmamau)

r = matrix(0,n,1)
x = matrix(0,n,20)

aux=propbons*n
for(i in 1:aux)
{
r[i] = 0
x[i,]=bons2[i,]
}
aux=propbons*n + 1
for(i in aux:n)
{
r[i] = 1
x[i,]=maus2[i-aux+1,]
}

dados = data.frame(x)

pdct = predict(modelo, newdata=dados, type="response")

r=factor(r)
vauc[j]=auc(roc(pdct,r))

for(i in 1:n)
{
if (pdct[i]>0.3)
{
pdct[i]=1
}
else {
pdct[i] = 0
}
}


acc=0
mM=0
```

```
bB=0
M=0 ##Quantidade de inadimplentes
B=0 ##Quantidade de adimplentes

for(i in 1:n)
{
if (r[i]==pdct[i])
{
acc = acc + 1
}
if (r[i]==1 && pdct[i]==1)
{
mM = mM + 1
}
if (r[i]==0 && pdct[i]==0)
{
bB = bB + 1
}
if (r[i]==1)
{
M = M + 1
}
if (r[i]==0)
{
B = B + 1
}
}
vacc[j]=acc/n
vmM[j] = mM/M
vbB[j]=bB/B

}

write("Simulacao maus- bons(10-90 - logistico)","C:/Users/Renato/Desktop/codigos dissertacao/dados_logistico_10_90.txt", append = TRUE)

write(" ","C:/Users/Renato/Desktop/codigos dissertacao/dados_logistico_10_90.txt", append = TRUE)
write(" ","C:/Users/Renato/Desktop/codigos dissertacao/dados_logistico_10_90.txt", append = TRUE)

write("##############--ACC--#################","C:/Users/Renato/Desktop/codigos dissertacao/dados_logistico_10_90.txt", append = TRUE)

write(vacc,"C:/Users/Renato/Desktop/codigos dissertacao/dados_logistico_10_90.txt", append = TRUE)

write("##############--bB--#################","C:/Users/Renato/Desktop/codigos dissertacao/dados_logistico_10_90.txt", append = TRUE)

write(vbB,"C:/Users/Renato/Desktop/codigos dissertacao/dados_logistico_10_90.txt", append = TRUE)

write("##############--mM--#################","C:/Users/Renato/Desktop/codigos dissertacao/dados_logistico_10_90.txt", append = TRUE)

write(vmM,"C:/Users/Renato/Desktop/codigos dissertacao/dados_logistico_10_90.txt", append = TRUE)

write("##############--auc--#################","C:/Users/Renato/Desktop/codigos dissertacao/dados_logistico_10_90.txt", append = TRUE)

write(vauc,"C:/Users/Renato/Desktop/codigos dissertacao/dados_logistico_10_90.txt", append = TRUE)
```

Código para análise de discriminante quadrática (QDA).

```
library(e1071)
library("MASS")
library(AUC)

mubom = c(0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)

mumau = c(1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),
1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20),1/sqrt(20))

sigmabom = matrix(0,20,20)
sigmamau = matrix(0,20,20)

sigmabom[1,1]=4
sigmabom[2,2]=4
sigmabom[3,3]=4
sigmabom[4,4]=4
sigmabom[5,5]=4
sigmabom[6,6]=4
sigmabom[7,7]=4
sigmabom[8,8]=4
sigmabom[9,9]=4
sigmabom[10,10]=4
sigmabom[11,11]=4
sigmabom[12,12]=4
sigmabom[13,13]=4
sigmabom[14,14]=4
sigmabom[15,15]=4
sigmabom[16,16]=4
sigmabom[17,17]=4
sigmabom[18,18]=4
sigmabom[19,19]=4
sigmabom[20,20]=4


sigmamau[1,1]=1
sigmamau[2,2]=1
sigmamau[3,3]=1
sigmamau[4,4]=1
sigmamau[5,5]=1
sigmamau[6,6]=1
sigmamau[7,7]=1
sigmamau[8,8]=1
sigmamau[9,9]=1
sigmamau[10,10]=1
sigmamau[11,11]=1
sigmamau[12,12]=1
sigmamau[13,13]=1
sigmamau[14,14]=1
sigmamau[15,15]=1
sigmamau[16,16]=1
sigmamau[17,17]=1
sigmamau[18,18]=1
sigmamau[19,19]=1
sigmamau[20,20]=1

vacc = matrix(0,100)
vbB  = matrix(0,100)
vmM  = matrix(0,100)
vauc = matrix(0,100)

for (j in 1:100)
{
n=100000

r = matrix(0,n,1)
```

```
x=matrix(0,n,20)

propbons=0.9
propmaus=0.1

bons=mvrnorm(propbons*n,mubom, sigmabom)
maus=mvrnorm(propmaus*n,mumau, sigmamau)

aux=propbons*n
for(i in 1:aux)
{
r[i] = 0
x[i,]=bons[i,]
}
aux=propbons*n + 1
for(i in aux:n)
{
r[i] = 1
x[i,]=maus[i-aux+1,]
}


####################--QDA Treinamento---######################
r=factor(r)
dados = data.frame(r,x)

modelo.lda = qda(r ~ X1 + X2 + X3 + X4 + X5 + X6 + X7 + X8 + X9 + X10 + X11 + X12 + X13 + X14 + X15 + X16 + X17 + X18 + X19 + X20, data = dados)


############################################################

n = 20000
pdct=matrix(0,n)

propbons = 0.5
propmaus = 0.5

bons2=mvrnorm(propbons*n,mubom, sigmabom)
maus2=mvrnorm(propmaus*n,mumau, sigmamau)

r = matrix(0,n,1)
x = matrix(0,n,20)

aux=propbons*n
for(i in 1:aux)
{
r[i] = 0
x[i,]=bons2[i,]
}
aux=propbons*n + 1
for(i in aux:n)
{
r[i] = 1
x[i,]=maus2[i-aux+1,]
}

dados = data.frame(x)

   pdctAux=predict(modelo.lda, newdata=dados)
pdctAux=pdctAux$posterior

for(i in 1:n)
{
if (pdctAux[i,1] > pdctAux[i,2])
{
pdct[i]=0
} else
{
```

```
pdct[i]=1
}
}

acc=0
mM=0
bB=0
M=0 ##Quantidade de inadimplentes
B=0 ##Quantidade de adimplentes

for(i in 1:n)
{
if (r[i]==pdct[i])
{
acc = acc + 1
}
if (r[i]==1 && pdct[i]==1)
{
mM = mM + 1
}
if (r[i]==0 && pdct[i]==0)
{
bB = bB + 1
}
if (r[i]==1)
{
M = M + 1
}
if (r[i]==0)
{
B = B + 1
}
}
vacc[j]=acc/n
vmM[j] = mM/M
vbB[j]=bB/B
    r=factor(r)
vauc[j]=auc(roc(pdctAux[,2],r))
}


write("Simulacao maus- bons(10-90 - QDA)","C:/Users/Renato/Desktop/codigos dissertacao/dados_QDA_10_90.txt", append = TRUE)

write(" ","C:/Users/Renato/Desktop/codigos dissertacao/dados_QDA_10_90.txt", append = TRUE)
write(" ","C:/Users/Renato/Desktop/codigos dissertacao/dados_QDA_10_90.txt", append = TRUE)


write("##############--ACC--##################","C:/Users/Renato/Desktop/codigos dissertacao/dados_QDA_10_90.txt", append = TRUE)
write(vacc,"C:/Users/Renato/Desktop/codigos dissertacao/dados_QDA_10_90.txt", append = TRUE)

write("##############--bB--##################","C:/Users/Renato/Desktop/codigos dissertacao/dados_QDA_10_90.txt", append = TRUE)
write(vbB,"C:/Users/Renato/Desktop/codigos dissertacao/dados_QDA_10_90.txt", append = TRUE)

write("##############--mM--##################","C:/Users/Renato/Desktop/codigos dissertacao/dados_QDA_10_90.txt", append = TRUE)
write(vmM,"C:/Users/Renato/Desktop/codigos dissertacao/dados_QDA_10_90.txt", append = TRUE)

write("##############--AUC--##################","C:/Users/Renato/Desktop/codigos dissertacao/dados_QDA_10_90.txt", append = TRUE)
write(vauc,"C:/Users/Renato/Desktop/codigos dissertacao/dados_QDA_10_90.txt", append = TRUE)
```