Universidade Federal do Piauí
Centro de Ciências da Natureza
Programa de Pós-Graduação em Ciência da Computação

Investigação dos Efeitos do Desbalanceamento de Classes na Aprendizagem da Regulação de Planos de Saúde

Jackson Cunha Cassimiro

Número de Ordem PPGCC: M001

Teresina-PI, 16 de setembro de 2016

Jackson Cunha Cassimiro

Investigação dos Efeitos do Desbalanceamento de Classes na Aprendizagem da Regulação de Planos de Saúde

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da UFPI (área de concentração: Sistemas de Computação), como parte dos requisitos necessários para a obtenção do Título de Mestre em Ciência da Computação.

Universidade Federal do Piauí – UFPI

Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação

Orientador: André Macedo Santana

Teresina-PI

16 de setembro de 2016

Jackson Cunha Cassimiro
Investigação dos Efeitos do Desbalanceamento
de Classes na Aprendizagem da Regulação de Planos de Saúde/ Jackson Cunha Cassimiro. – Teresina-PI, 16 de setembro de 2016-
66 p. : il. (algumas color.) ; 30 cm.

Orientador: André Macedo Santana

Dissertação (Mestrado) – Universidade Federal do Piauí – UFPI
Centro de Ciências da Natureza
Programa de Pós-Graduação em Ciência da Computação, 16 de setembro de 2016.

1. Mineração de Dados. 2. Aprendizagem de Máquina.
CDU 02:141:005.7

Investigação dos Efeitos do Desbalanceamento de Classes na Aprendizagem da Regulação de Planos de Saúde

JACKSON CUNHA CASSIMIRO

Dissertação apresentada ao Programa de Pós–Graduação em Ciência da Computação do Centro de Ciências da Natureza da Universidade Federal do Piauí, como parte integrante dos requisitos necessários para obtenção do grau de Mestre em Ciência da Computação.

Aprovado por:

Prof. André Macêdo Santana

(Presidente da Banca Examinadora)

Prof. Cleber Zanchettin

(Examinador Externo)

Prof. Hermes Manoel Galvão Castelo Branco

(Examinador Externo)

Prof. Pedro de Alcântara dos Santos Neto

(Examinador Interno)

Teresina, 16 de setembro de 2016

À minha mãe,
por todo o amor e dedicação.

Agradecimentos

Agradeço a minha mãe, Maria das Graças, pelo amor de todos os dias.

À minha família, por todo o apoio.

À Érica Lays, pelo companheirismo e paciência.

Aos meus amigos, pelas conversas e esclarecimentos.

Agradeço aos meus orientadores, André Macedo e Pedro de Alcântara, pelo acompanhamento e orientações.

À Infoway, pelo apoio desprendido para a realização deste trabalho de pesquisa.

"Quem se fortalece por todos os lados,
Se enfraquece por todos os lados.
(Sun Tzu)

Resumo

A operação de planos privados de assistência à saúde no Brasil representa uma importante via de prestação de serviços à população. O Brasil é o maior mercado de saúde privada na América do Sul, sendo que em 2012 os custos com saúde representaram cerca de 8% do PIB. Nesse mercado, muitas empresas operadoras de planos de saúde (OPS) encontram-se em situação de desequilíbrio financeiro, caracterizada pelo fato de as despesas somadas representarem um valor maior que as receitas. Fraudes e abusos na utilização de serviços em saúde são dois fatores que influenciam diretamente esse desequilíbrio, uma vez que correspondem a despesas que poderiam ser eliminadas sem prejuízo à qualidade dos serviços prestados. Um dos mecanismos empregados pelas OPS para evitar despesas indevidas decorrentes de fraudes e abusos é a Regulação, que consiste em uma análise prévia antes da liberação para realização, dos serviços que são solicitados pelos prestadores de saúde. A análise manual das solicitações que é realizada durante a regulação de planos de saúde é um exemplo de fator que tem motivado as OPS a desenvolverem sistemas capazes de identificar fraudes e abusos de forma automática ou semi-automática, muitas vezes por meio de técnicas de Mineração de Dados e Aprendizagem de Máquina. Neste cenário, a utilização dessas técnicas é impactada pelo problema do desbalanceamento de classes, oriundo do fato de haver muito mais solicitações de serviços autorizadas do que não autorizadas pelo processo de regulação. A proposta deste trabalho é investigar os efeitos desse problema na aplicação de técnicas de aprendizagem de máquina no contexto da regulação de planos de saúde. Mais precisamente, é investigar por meio de um experimento o quanto de performance de predição é perdida devido ao desbalanceamento de classes e o quanto dessa performance perdida pode ser recuperada utilizando-se métodos de tratamento específicos aplicados aos dados. Este experimento emprega bases de dados em que as distribuições de classes foram modificadas artificialmente, algoritmos de classificação de diferentes paradigmas e diferentes métodos de tratamento de dados. Entre os resultados mais importantes, notou-se que o desbalanceamento de classes afeta sim a performance de aprendizagem da regulação, mas de forma diferente para cada algoritmo estudado. Observou-se também que os métodos de tratamento são capazes de reduzir a perda de performance, mas também que essa redução depende do algoritmo de classificação e da distribuição de classes empregados em conjunto.

Palavras-chaves: aprendizagem de máquina, mineração de dados, balanceamento de classes, planos de saúde, regulação de planos de saúde.

Abstract

Private health insurance services in Brazil are an important way of providing health to population. Brazil is the largest private healthcare market in South America, and in 2012 health care costs accounted for about 8 % of GDP. In Brazil many health insurance companies (HIC) are in financial imbalance, in which the added costs are greater than revenues. Fraud and abuse in consuption of healthcare are two factors that directly influence the costs, since they correspond to expenses that could be eliminated without prejudice to the quality of services provided. One of the mechanisms employed by HIC to avoid undue expenses caused by fraud and abuse is a claim authorization process, which consists of a preliminary analysis before release to execution. The manual analysis of claims performed is a factor that has motivated HIC to develop systems able to identify claims linked to fraud and abuse in an automatic or semi-automatic manner, often using data mining and machine learning techniques due to the large amount of data produced by these systems. The use of these techniques is affected by the problem of class imbalance, arising from the fact that the claim authorization process produces more authorized claims than not authorized ones. The purpose of this study is to investigate the effects of class imbalance in the claim authorization domain. More precisely, the goal is to investigate by an experiment how prediction performance is lost due to class imbalance and how much can be recovered using specific treatment methods applied to the data. This experiment employs databases in which class distributions have been modified artificially, classification algorithms of different paradigms and different treatment methods. Among the most important results, it was noted that the class imbalance does affect the performance of the claim authorization learning, but differently for each studied algorithm. It was also observed that treatment methods can reduce loss of performance, but also that this reduction depends on the classification algorithm and class distribution used together.

Keywords: machine learning, data mining, class imbalance, health insurance, claim authorization.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ANS	*Agência Nacional de Saúde Suplementar*
API	*Application Program Interface*
mRMR	*minimum Redundancy Maximum Relevance*
OPS	*Operadora de Planos de Saúde*
PIB	*Produto Interno Bruto*
RGI	*Razão de Ganho de Informação*
RIPPER	*Repeated Incremental Pruning to Produce Error Reduction*
SMOTE	*Synthetic Minority Over-sampling Technique*
WEKA	*Waikato Environment for Knowledge Analysis*

Sumário

1 Introdução

Este é o capítulo que apresenta em linhas gerais todo o conteúdo deste trabalho. No início é apresentada uma contextualização sobre o mercado de planos de saúde no Brasil e em seguinda defini-se o problema que será abordado. Em seguida, é apresentada a proposta juntamente com os objetivos que podem ser atingidos. Por fim, são apresentadas as justificativas para o desenvolvimento deste trabalho bem como as suas principais contribuições.

1.1 Contexto e Motivação

A operação de planos privados de assistência à saúde no Brasil representa uma importante via de prestação de serviços à população. Na América do Sul, o Brasil é o maior mercado de saúde privada. Em 2012 os custos com saúde representaram cerca de 8% do PIB (HILLERMAN; CARVALHO; REIS, 2015). Segundo dados da Agência Nacional de Saúde Suplementar - ANS, órgão governamental que regula o setor de planos de saúde privados no Brasil, até setembro de 2015 havia cerca de 50.261.602 beneficiários [1], conforme mostra a Figura 1 (ANS, 2015):

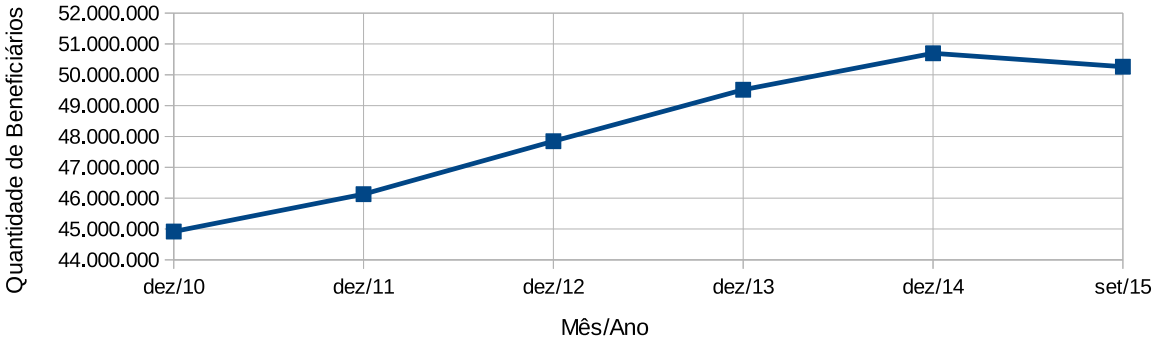

Figura 1 – Quantidade de beneficiários de planos de saúde no Brasil.

Uma característica importante desse mercado com implicações diretas para as operadoras de planos de saúde é a relação entre receitas e despesas. Segundo informações disponibilizadas pela ANS e resumidas na Figura 2, muitas operadoras encontram-se em situação de desequilíbrio financeiro, caracterizada pelo fato das despesas representarem um valor maior que as receitas.

A situação apresentada na Figura 2 pode ficar ainda pior, seguindo a tendência mundial de aumento dos custos com saúde que, entre outros fatores, é influenciada pelo

[1] O termo "beneficiário" refere-se a vínculos aos planos de saúde, podendo incluir vários vínculos para um mesmo indivíduo.

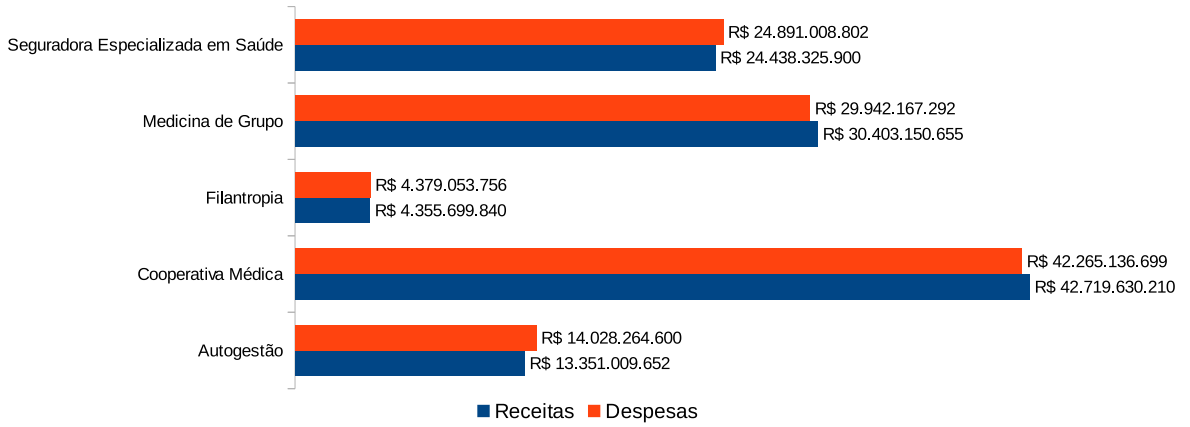

Figura 2 – Comparação entre receitas e despesas das operadoras de planos de saúde do Brasil, agrupadas por tipo de operadora (dezembro, 2015).

aumento da expectativa de vida e avanços na área de saúde (KOSE; GOKTURK; KILIC, 2015). Como exemplo dessa tendência, pode-se citar a evolução dos gastos com saúde nos Estados Unidos que, segundo estimativas, no ano 2000 correspondeu a 13.8% do PIB (KOSE; GOKTURK; KILIC, 2015), no ano 2008 aumentou para 15.2% do PIB (CHANDOLA; SUKUMAR; SCHRYVER, 2013) e em 2016 poderá alcançará a marca de 19.6% do PIB (DUA; BAIS, 2014).

Outros dois fatores importantes que influenciam diretamente os custos com saúde são fraudes e abusos, uma vez que correspondem a despesas que poderiam ser eliminadas sem prejuízo na qualidade dos serviços prestados (KELLEY, 2009). Entende-se por fraude a produção intencional de informações falsas por uma entidade ou indivíduo, sabendo-se que essas informações falsas resultarão em algum benefício para essa entidade, indivíduo ou terceiros. Já abusos, no âmbito da assistência à saúde, podem ser entendidos como práticas que são inconsistentes com os critérios médicos e administrativos pré-estabelecidos (KOSE; GOKTURK; KILIC, 2015).

Estima-se que nos Estados Unidos, cerca de 3% a 10% dos gastos com saúde dos setores público e privado estão relacionados a fraudes, o que representou no ano fiscal de 2009 gastos entre 75 e 250 bilhões de dólares (MORRIS, 2009). Além das perdas financeiras, há também a perda de qualidade da assistência prestada aos beneficiários, seja devido a escassez de recursos ou pela exposição a tratamentos de risco desnecessários (DUA; BAIS, 2014; KELLEY, 2009). Não foram encontradas estimativas semelhantes para o mercado de saúde brasileiro, no entanto acredita-se que, de forma análoga, fraudes e abusos também representam uma parcela importante dos gastos com saúde.

O controle eficiente da detecção de fraudes e abusos é crucial para as operadoras de planos de saúde que, como mostra a Figura 2, precisam reduzir os custos até níveis estáveis enquanto oferecem serviços adequados de assistência a saúde. Por sua vez, as operadoras tem encontrado dificuldades para descobrir comportamentos suspeitos, devido principalmente ao

enorme volume de informações que precisam ser processadas (HILLERMAN; CARVALHO; REIS, 2015).

As operadoras de planos de saúde dispõem de diversos mecanismos para controle dos custos assistenciais e evitar despesas indevidas decorrentes de fraudes ou abusos. Um deles é a Regulação, que consiste em uma análise prévia dos serviços que são solicitados pelos prestadores de saúde (clínicas, hospitais, etc.) em função do atendimento prestado aos beneficiários.

A regulação possui relação direta com os custos assistenciais e administrativos. Para que sua implantação seja efetiva, é preciso que haja uma equipe de profissionais dedicados à tarefa de analisar as solicitações de serviços, o que pode encarecer os custos administrativos principalmente para as operadoras de menor porte. Já a não implantação da regulação permite que muitas solicitações indevidas e até mesmo fraudulentas sejam realizadas, aumentando os custos assistenciais.

Atualmente há uma demanda por sistemas capazes de, automaticamente, acompanhar o comportamento de prestadores de serviço e permitir a identificação de comportamentos suspeitos e possivelmente fraudulentos (HILLERMAN; CARVALHO; REIS, 2015). A análise manual das solicitações que comumente é realizada durante a regulação de planos de saúde é um exemplo de fator que tem motivado as operadoras de planos de saúde a desenvolverem sistemas capazes de identificar fraudes e abusos de forma automatizada (KOSE; GOKTURK; KILIC, 2015). Neste contexto, técnicas de mineração de dados tem sido empregadas para prover informações capazes de melhorar a eficiência, reduzir os custos e aumentar os rendimentos enquanto preserva-se o alto nível da qualidade dos serviços assistenciais (DUA; BAIS, 2014).

1.2 Definição do Problema

A aplicação de técnicas de mineração de dados tem se tornado comum na detecção de fraudes e abusos em sistemas de saúde, principalmente devido a grande quantidade de dados produzidos nesses sistemas e a inviabilidade de processamento por meios tradicionais (DUA; BAIS, 2014). Entre as técnicas de mineração de dados mais utilizadas destaca-se aquelas de aprendizagem supervisionada, que podem ser empregadas para classificar as operações entre fraudulentas ou não.

No que diz respeito a detecção de fraudes e abusos na operação de planos de saúde, técnicas de aprendizagem supervisionada podem ser aplicadas para realizar a classificação de serviços solicitados durante a regulação. Sob essa abordagem, os dados das solicitações que já foram analisadas anteriormente são empregados na construção de classificadores que posteriormente classificarão automaticamente com uma dada taxa de acerto as solicitações entre autorizadas e não autorizadas, deixando que apenas as solicitações com maiores

indícios de fraudes sejam analisadas por um profissional humano. No entanto, a construção desses classificadores é comprometida devido a características inerentes aos dados utilizados no treinamento (STEFANOWSKI, 2013). No caso da regulação de planos de saúde, uma dessas características é o desbalanceamento de classes.

No âmbito da aprendizagem de máquina, uma base de dados é dita desbalanceada quando há muito menos exemplares de uma das classes (STEFANOWSKI, 2013). Tomando-se como exemplo um problema envolvendo duas classes (classificação binária), como é o caso da regulação de planos de saúde, a classe que contém menos exemplares é dita minoritária enquanto que a outra classe é chamada de majoritária. Embora seja de amplo conhecimento que o desbalanceamento de classes afeta a aprendizagem de algoritmos de classificação, não se sabe a extensão desse problema no contexto da regulação de planos de saúde, em que é comum haver muito mais solicitações autorizadas do que não autorizadas.

Diante dessa situação, o problema abordado nesse trabalho é a investigação dos efeitos do desbalanceamento de classes quanto a aprendizagem da regulação de planos de saúde. Mais precisamente, deseja-se investigar o quanto da performance de aprendizagem é perdida devido ao desbalanceamento de classes e também o quanto é possível recuperar por meio de métodos de tratamento para o desbalanceamento de classes.

1.3 Visão Geral da Proposta

A proposta deste trabalho é investigar os efeitos do desbalanceamento de classes seguindo-se o experimento proposto por Prati, Gustavo E A P e Silva (2014). Por meio desse experimento, diversos classificadores são construídos utilizando-se bases de dados cujas distribuições de classes foram modificadas artificialmente e também bases de dados tratadas para corrigir o desbalanceamento de classes. Esse experimento permite medir a performance de predição dos classificadores antes e depois da utilização dos métodos de tratamento e com isso obter diversas informações úteis sobre como o desbalanceamento de classes afeta a aprendizagem dos algoritmos no domínio da regulação de planos de saúde. Para maior clareza, esse experimento foi dividido em 4 etapas:

- Divisão das bases de dados: nesta etapa cada uma das bases de dados empregadas no experimento são divididas em bases menores destinadas ao treinamento e teste dos algoritmos de classificação. As bases de teste possuem a mesma distribuição de classes da base de dados da qual foi originada, já as bases de dados destinadas ao treinamento possuem diferentes distribuições de classes;

- Medição da perda de performance: diversos classificadores são construídos utilizando-se as bases de dados de diferentes distribuições de classes e em seguida as medidas de performance são obtidas utilizando-se a base de dados destinada aos testes;

- Medição da recuperação de performance: nesta etapa as mesmas bases de dados empregadas na etapa anterior são tratadas com métodos de tratamento com o intuito de corrigir o desbalanceamento de classes. Em seguida, são construídos novos classificadores utilizando-se as bases de dados tratadas e então as performances de predição são novamente mensuradas;
- Análise dos resultados: nesta etapa as medidas de performance resumidas em médias e intervalos de confiança e também agrupadas por algoritmo, método de tratamento e distribuição de classes para uma melhor compreensão.

1.4 Objetivos

O objetivo principal deste trabalho é investigar o quanto da performance de predição é perdida devido ao desbalanceamento de classes e o quanto pode ser recuperada por meio da utilização de métodos de tratamento no âmbito da regulação de planos de saúde.

Juntamente com este objetivo principal, pretende-se atingir outros objetivos específicos, que são:

- investigar como varia a performance de predição de diversos classificadores em diferentes distribuições de classes;
- investigar, entre os algoritmos de classificação empregados, qual deles é mais ou menos afetado pelo desbalanceamento de classes;
- investigar o quanto da performance perdida pode ser recuperada por meio da utilização métodos de tratamento;
- investigar quais combinação de algoritmos de classificação e método de tratamento pode ser mais benéfica para a aprendizagem da regulação de planos de saúde.

1.5 Justificativa

As operadoras de planos de saúde no Brasil, tanto as que atuam no setor privado quanto aquelas que atuam no setor público, encontram dificuldades em detectar transações relacionadas a fraudes e abusos. O processo de regulação, no qual os serviços solicitados são submetidos a uma análise prévia antes de serem autorizados pelas operadoras, é uma das formas de detectar esse tipo de transação. No entanto, esse processo é prejudicado devido ao grande volume de informações que precisa ser analisado, muitas vezes manualmente (HILLERMAN; CARVALHO; REIS, 2015).

O problema decorrente dessa análise manual pode ser atenuado explorando-se formas de classificar automaticamente as solicitações de serviços. Por essa abordagem, os

dados históricos das solicitações que já passaram pela regulação podem ser empregados no treinamento de algoritmos de classificação. O resultado desse treinamento é a construção de modelos capazes de classificar automaticamente, com uma certa taxa de erro, novas solicitações de serviços. Embora se tenha conhecimento de trabalhos que explorem a performance desse tipo de algoritmo em bases de dados sobre regulação de planos de saúde (ARAÚJO; SANTANA; NETO, 2015), não se sabe de nenhum estudo sobre como as características intrínsecas dos dados sobre regulação influenciam a performance da aprendizagem alcançada.

Uma dessas características é o desbalanceamento de classes, que pode ser observada em bases de dados sobre regulação devido ao fato de haver muito mais solicitações de serviços autorizadas do que não autorizadas. Nessas circunstâncias, a aprendizagem dos algoritmos é afetada e comumente os classificadores obtidos são enviesados, obtendo maiores performances no reconhecimento da classe majoritária do que da minoritária (STEFANOWSKI, 2013).

Pelo que foi exposto acima, percebe-se a importância da compreensão de como as características intrínsecas dos dados influencia a aprendizagem da regulação de planos de saúde, entre elas o desbalanceamento de classes. Esta compreensão serve de suporte para o desenvolvimento de sistemas inteligentes que apoiem a detecção de fraudes e abusos no mercado de planos de saúde.

1.6 Contribuições

As principais contribuições deste trabalho são:

- O desenvolvimento de um relato mostrando como diversos algoritmos de classificação são afetados pelo desbalanceamento de classes, em diversas distribuições de classes diferentes. Com essa informação é possível notar que alguns algoritmos, e provavelmente outros do mesmo paradigma, são mais ou menos afetados pelo desbalanceamento de classes;

- A exibição de resultados experimentais que apresentam quanto da performance perdida pode ser recuperada por meio da utilização de um determinado método de tratamento aplicado aos dados de treinamento. Essa informação ainda pode ser detalhada por distribuição de classes, permitindo comparar como cada método se comportou em uma dada distribuição ou mesmo por algoritmo de classificação, permitindo identificar se uma determinada combinação de algoritmo de classificação e método de tratamento é ou não benéfica;

- A descoberta de como as características intrínsecas de cada método de tratamento somadas às características dos dados de regulação de planos de saúde pode influenciar

a performance dos algoritmos de classificação.

1.7 Estrutura do Trabalho

Os demais capítulos deste trabalho estão organizados da seguinte forma: o Capítulo 2 comenta sobre trabalhos que exploram o problema da aprendizagem em bases de dados desbalanceadas; o Capítulo 3 resume alguns tópicos sobre Mineração de Dados e Aprendizagem de Máquina que são úteis para a compreensão da proposta deste trabalho; o Capítulo 4 apresenta como o desbalanceamento de classes será investigado no contexto da regulação de planos de saúde; o Capítulo 5 apresenta os resultados da aplicação do experimento descrito no Capítulo 4; por fim, o Capítulo 6 conclui este trabalho e comenta também sobre possíveis direcionamentos para trabalhos futuros. Há também o Apêndice A, que resume como uma das bases de dados utilizada neste trabalho foi concebida.

2 Trabalhos Relacionados

Este capítulo apresenta uma visão geral sobre o problema de desbalanceamento de classes, mostrando alguns conceitos importantes e como os trabalhos nessa área estão organizados. A Seção 2.2 comenta sobre outros estudos que procuram responder questões importantes sobre a relação entre distribuição de classes e performance de classificadores. Por fim, a Seção 2.3 apresenta algumas soluções desenvolvidas em domínios específicos, procurando descrever resumidamente como o estudo foi conduzido, que algoritmos e métodos de tratamento de dados foram implementados e os principais resultados.

2.1 Estudos sobre Desbalanceamento de Classes

Os estudos que tratam sobre o tema de desbalanceamento de classes comumente podem ser agrupados em três categorias (GARCÍA; SáNCHEZ; MOLLINEDA, 2012):

- Aqueles focados na implementação de soluções para o desbalanceamento de classes em nível de algoritmos ou dos dados;
- Aqueles que procuram medir a performance dos classificadores nos diversos domínios em que ocorre esse problema;
- Aqueles que investigam outros fatores que podem estar relacionados a performance dos classificadores em domínios específicos em que os dados encontram-se desbalanceados.

Este trabalho pode ser enquadrado no segundo grupo, visto que procura estudar a relação entre performance dos classificadores e distribuição de classes no domínio da regulação de planos de saúde.

2.2 Distribuição de Classes e Performance dos Classificadores

O conjunto de experimentos realizados neste trabalho sobre os efeitos do desbalanceamento de classes na aprendizagem da regulação de planos de saúde foi inspirado no trabalho de Prati, Gustavo E A P e Silva (2014). No trabalho citado é proposto um experimento para avaliar a influência do desbalanceamento de classes na performance de classificadores e também foi proposto um procedimento estatístico que faz uso de intervalos de confiança para apoiar as conclusões. O experimento descrito emprega vinte e duas bases de dados, diversos tipos de algoritmos de classificação e métodos de tratamento. As medições de perda e recuperação de performance são calculadas de forma relativa,

considerando a performance na base de dados perfeitamente balanceada (50/50). Os resultados obtidos nesse estudo indicam que a perda de performance alcança valores em torno de 5% quando a classe minoritária representa 10% ou mais do total de exemplares na base de dados. Para os níveis de balanceamento mais extremos, a perda de performance chega a 20%. Quanto a performance que pode ser recuperada pela utilização de métodos de tratamento, esse estudo mostra que, em média, apenas cerca de 30% da performance perdida pode ser recuperada pelos métodos de tratamento estudados.

Uma etapa importante do experimento citado no parágrafo anterior é a construção de diversas bases de dados com a quantidade de exemplares fixa e a distribuição de classes modificada artificialmente, para que então sejam utilizados os algoritmos de classificação e métodos de tratamento para desbalanceamento. Essa ideia é inspirada no trabalho de Weiss e Provost (2003), que procurou responder a seguinte pergunta: se apenas n exemplares podem ser obtidos para o treinamento, em que proporção as classes devem estar distribuídas? Essa pergunta está relacionada a domínios em que a obtenção de dados para o treinamento de algoritmos é muito custosa e faz-se necessário limitar o volume de dados utilizado para o treinamento. Nesse estudo são utilizadas vinte e seis bases de dados e apenas um único algoritmo de classificação: C4.5. Os resultados encontrados indicam que, comumente, para cada base de dados há uma distribuição de classes diferente da natural (ou original) que permite a obtenção de classificadores com melhores performances.

Albisua et al. (2012) também investigaram a relação entre o balanceamento de classes e a performance de classificadores, procurando responder as seguintes perguntas:

- Será a distribuição de classes 50/50 a mais indicada para o treinamento de classificadores?
- Uma distribuição de classes ótima é dependente dos métodos de *resampling* e algoritmos de classificação utilizados?
- Há vantagens em utilizar métodos de *resampling*, mesmo para aqueles domínios em que as bases de dados já encontra-se balanceadas?

Albisua et al. (2012) apresentam uma metodologia para encontrar uma distribuição de classes quase ótima. Nos experimentos utilizaram 29 bases de dados, 8 diferentes métodos de *resampling* e dois algoritmos de classificação (C4.5 e PART). Na avaliação da performance são utilizados a métrica AUC e testes estatísticos. Os resultados confirmam que a distribuição de classes ótima depende do domínio, do algoritmo de classificação e do método de *resampling* utilizado.

García, Sánchez e Mollineda (2012) estudaram o quanto a efetividade dos métodos de tratamento de dados baseados em *undersampling* e *oversampling* é afetada pelo desbalanceamento de classes ou pelos algoritmos de classificação. Nesse trabalho foram empregadas

17 bases de dados reais, 8 algoritmos de classificação, 2 métodos de *oversampling*, 2 métodos de *undersampling* e 4 métricas para avaliar a performance. Como o principal interesse é comparar os métodos de *oversampling* e *undersampling*, as medidas de performance foram separadas em dois grupos e utilizadas no cálculo de médias separadamente. Os resultados obtidos indicam que os métodos de *oversampling* foram superiores aos de *undersampling* nas bases de dados cujo desbalanceamento é alto ou moderado. Já nas bases de dados em que o desbalanceamento é baixo, a diferença de performance não foi significante. Outro resultado importante desse estudo é que os algoritmos de classificação tiveram pouca influência na eficácia dos métodos de tratamento estudados.

Seiffert et al. (2014) apresentam um conjunto de experimentos que investigam os impactos causados pelo desbalanceamento de classes e ruídos na performance de classificadores destinados a identificar módulos com possíveis falhas em software. O trabalho desses autores se assemelha a este, que investiga o desbalanceamento de classes no contexto da regulação de planos de saúde, principalmente devido às questões que eles pretendem esclarecer:

- Qual é o impacto relativo do ruído de classe versus desbalanceamento de classes? Qual é mais danoso à performance de diferentes algoritmos de classificação e métodos de *resampling*?
- Como a performance de diferentes algoritmos de classificação varia segundo a aplicação de técnicas de *resampling* (tratamento dos dados)?
- Quais algoritmos são mais beneficiados pelas técnicas de *resampling*?
- Alguma técnica de *resampling* funciona melhor com um algoritmo de classificação específico?
- Quais os benefícios proporcionados pelas técnicas de *resampling* em diferentes níveis de desbalanceamento de classes e ruídos? Se os dados encontram-se altamente desbalanceados ou com muito ruído, que técnicas de *resampling* implicam em melhores resultados?

No estudo citado cima, foram utilizados 11 algoritmos de classificação, 7 técnicas de *resampling* e 1 base de dados da qual foram derivadas outras 12, em que os níveis de desbalanceamento de classes e ruídos foram modificados artificialmente. Os resultados apontam que as técnicas *Wilson's Editing* e *Undersampling* Aleatório tiveram bons resultados. Entre os algoritmos de classificação, alguns foram mais beneficiados pelos métodos de *resampling* que outros. Outro resultado importante é que o ruído de classe, situação em que um exemplar da base de dados possui um valor incorreto de classificação, possui um impacto mais significante na performance dos classificadores do que o desbalanceamento.

2.3 Soluções em Domínios Específicos

Diferentes experimentos e propostas tem sido desenvolvidos para investigar ou contornar os desafios da aprendizagem em bases de dados desbalanceadas em domínios específicos. A seguir estão listados alguns desses trabalhos.

Xiao et al. (2012) apresentam um método para a aprendizagem em bases de dados desbalanceadas que combina *ensembles* e aprendizagem baseada em custos (*cost-sensitive learning*). Por esse método, cada exemplar da base de dados é classificado pelo *ensemble* mais apropriado selecionado dinamicamente. Os resultados desse experimento são demonstrados em bases de dados sobre análise de crédito e cancelamento de contratos em empresa de telecomunicações e apontam que o método proposto pode trazer melhores resultados que outros métodos também baseados em *ensembles*.

Das, Krishnan e Cook (2014) propõem uma nova técnica de preprocessamento chamada ClusBUS. Essa técnica utiliza um algoritmo de agrupamento baseado em densidade chamado DBSCAN para formar grupos de interesse que em seguida são submetidos a *undersampling*. Nesse estudo são utilizados dados gerados a partir de leituras de sensores que monitoram tarefas sendo executadas em uma residência. Segundo os autores, é possível obter resultados superiores comparando-se com técnicas de preprocessamento conhecidas, como o SMOTE.

Em seus estudos sobre a predição de defeitos em software, domínio em que também há desbalanceamento de classes, Siers e Islam (2015) propõem uma técnica de classificação baseada em custos chamada CSForest. Esta técnica consiste em um *ensemble* de árvores de decisão em que as saídas são combinadas por meio de votação utilizando um esquema baseado em custos chamado CSVoting, que também foi desenvolvido por esses autores. Nesse estudo, a técnica proposta é comparada com seis outros algoritmos de classificação utilizando-se seis bases de dados disponíveis publicamente sobre defeitos em software. Os resultados encontrados indicam que o método proposto foi superior aos utilizados na comparação.

Dittman, Khoshgoftaar e Napolitano (2015) procuraram determinar se a utilização de métodos de *undersampling* seria capaz de melhorar a performance do algoritmo *Random Forest* em bases de dados sobre bioinformática. Em seus experimentos foram utilizadas 15 bases de dados contendo dados sobre microarranjos de DNA. Cada uma dessas bases, que originalmente encontravam-se desbalanceadas, foram tratadas segundo o método *undersampling* aleatório para que atingissem a distribuição de classes final de 35/65 e 50/50. Em seguida, as bases de dados originais e tratadas foram utilizadas no treinamento de classificadores e a respectiva performance de classificação foi medida segundo a métrica AUC (área sob a curva ROC). Os resultados indicam que, embora a performance tenha melhorado pela utilização do método de *undersampling*, a diferença não foi estatisticamente

significativa. Logo, os autores concluíram que o algoritmo *Random Forest* é robusto o suficiente para ser aplicado em bases de dados sobre microarranjos de DNA, mesmo essas bases estando desbalanceadas.

3 Tópicos em Mineração de Dados e Aprendizagem de Máquina

Este capítulo discute sobre diversos tópicos que ajudam a compreender como o processo de regulação de planos de saúde pode ser melhorado, automatizando-se a tarefa de classificar as solicitações de serviço "autorizado" ou "não autorizado". Também comenta sobre balanceamento de classes e como esta característica dos dados influencia na aprendizagem da regulação. Há ainda descrições sobre os algoritmos de classificação e métodos de tratamento para bases de dados desbalanceadas empregados no experimento que será abordado nos capítulos seguintes.

3.1 Mineração de Dados

Mineração de dados é o processo de descoberta de padrões de forma automática ou semiautomática em uma quantidade suficiente de dados (WITTEN; FRANK; HALL, 2011; HAN; KAMBER; PEI, 2011). As fontes de dados para mineração podem ser de variados tipos, como bancos de dados relacionais, *data warehouses*, *web*, fluxos contínuos (*streams*) ou qualquer outro repositório de informações.

No âmbito da regulação de planos de saúde, a mineração de dados pode ser utilizada para descobrir padrões capazes de explicar o porquê de algumas solicitações de serviços de saúde serem autorizadas ou não. Esses padrões podem ser representados de diversas formas, sendo que algumas delas permitem que suas estruturas internas sejam analisadas e facilmente compreendidas. Exemplos dessas formas de representação são as árvores e regras de decisão.

A maioria das técnicas utilizadas no descobrimento desses padrões foram desenvolvidas em uma área chamada Aprendizagem de Máquina.

3.2 Aprendizagem de Máquina

A Aprendizagem de Máquina é uma disciplina focada em responder duas questões inter-relacionadas: como construir programas de computador capazes melhorarem a sí mesmos automaticamente por meio da experiência? E quais são os princípios fundamentais de estatística, computação e de teoria da informação que governam os sistemas com capacidades de aprendizagem como programas de computador, seres humanos e organizações? O estudo dessa área de conhecimento é importante tanto por procurar responder essas

questões quanto pelas soluções que tem sido produzidas em diversas áreas de aplicação (JORDAN; MITCHELL, 2015).

Em Aprendizagem de Máquina, um problema de aprendizagem pode ser definido como a procura por melhorar alguma medida de performance enquanto se executa uma determinada tarefa, por meio de algum tipo de treinamento (JORDAN; MITCHELL, 2015). Por exemplo, na aprendizagem da regulação, a tarefa consiste em classificar as solicitações de serviços em duas categorias: "autorizado" e "não autorizado". A medida de performance a ser melhorada pode ser a acurácia desse do *software* que realiza a classificação e o treinamento pode ser realizado utilizando-se o histórico das solicitações que já passaram pelo processo de regulação anteriormente. Outra medida de performance que pode ser empregada consiste em atribuir um valor de custo para cada serviço classificado, sendo esse valor maior quando um serviço "não autorizado" for classificado incorretamente como "autorizado".

O funcionamento das técnicas de aprendizagem de máquina pode ser entendido como uma busca no espaço de possíveis programas capazes de melhorar uma determinada medida de performance (JORDAN; MITCHELL, 2015; WITTEN; FRANK; HALL, 2011). A variedade de técnicas existentes deve-se, em parte, a forma como essa busca é conduzida (medidas de convergência, métodos evolutivos, etc.) e também como a forma como esses programas são representados (regras, árvores de decisão, funções matemáticas, etc.).

Os métodos estudados em Aprendizagem de Máquina podem ser divididos em dois tipos: supervisionados e não supervisionados (LIBBRECHT; NOBLE, 2015). Os métodos não supervisionados procuram descobrir o relacionamento entre os atributos que caracterizam os dados sem a utilização de um atributo alvo, como nos métodos supervisionados. Esses métodos não serão detalhados neste trabalho, no entanto recomenda-se a leitura de Hastie, Tibshirani e Friedman (2009) para um melhor entendimento.

Os métodos de aprendizagem supervisionada são aqueles que procuram descobrir o relacionamento entre um conjunto de atributos de entrada (também chamados de variáveis independentes ou vetor de características) e um atributo alvo (também chamado de variável dependente ou rótulo) (ROKACH; MAIMON, 2010b; ZHANG; ZHOU, 2014). O relacionamento descoberto é representado segundo uma estrutura específica de cada método e são chamados de modelos (ROKACH; MAIMON, 2010b). Esses métodos podem ser divididos em dois tipos (ROKACH; MAIMON, 2010b): classificação e regressão.

Os métodos de classificação procuram mapear os atributos de entrada em um atributo de saída que representa um conjunto pré-definido de classes. Quanto aos métodos de regressão, o atributo que representa a saída assume valores contínuos. Os métodos de aprendizagem supervisionada e mais especificamente os de classificação estão entre os mais estudados na área de aprendizagem de máquina (ROKACH; MAIMON, 2010b; ZHANG; ZHOU, 2014).

3.3 Classificação

Os problemas que se procura resolver por meio da classificação podem ser definidos como buscar por uma função capaz de mapear um conjunto de exemplares em um conjunto pré-definido de rótulos ou classes com o menor erro de generalização (ROKACH; MAIMON, 2010b).

De forma simplificada, a resolução de um problema por meio de técnicas ou algoritmos de classificação pode ser dividida em duas fases, uma de treinamento e outra de classificação (HAN; KAMBER; PEI, 2011).

Durante a fase de treinamento, um modelo de classificação ou classificador é construído por meio da aplicação de um algoritmo de classificação à um conjunto de tuplas associadas a rótulos. Cada tupla, X, é representada por um vetor de n dimensões da forma $X = (x_1, x_2, ..., x_n)$. Cada elemento x_i corresponde ao valor de um atributo A_i que define as características da tupla. Um desses atributos é o atributo de classe, cujos valores são os rótulos associados a cada tupla e determinam a que categoria ou classe cada tupla pertence. No contexto da classificação essas tuplas também podem ser chamadas de exemplares ou instâncias.

Já na fase de classificação, também chamada de teste, o modelo construído na fase de treinamento é então utilizado para rotular tuplas ainda não rotuladas. O resultado obtido quando um modelo classifica uma tupla pode ser de dois tipos (AGGARWAL, 2014a):

- Rótulo: que representa a classe a qual pertence a tupla;
- Pontuação Numérica: para cada classe possível é retornada uma pontuação numérica que indica a propensão de a tupla pertencer a esta classe. Neste caso, a classificação final pode ser obtida escolhendo-se a classe que obtiver a maior pontuação numérica.

3.4 Avaliação de Performance

A avaliação da performance de classificadores quanto a predição é um assunto importante no âmbito da aprendizagem de máquina e da mineração de dados, visto que comumente essas medidas são utilizadas como principais indicadores da qualidade (PRATI; BATISTA; MONARD, 2011). Outros critérios não relacionados à predição e que também são utilizados na comparação de classificadores são (HAN; KAMBER; PEI, 2011):

- Velocidade: diz respeito aos custos computacionais envolvidos na construção e uso do classificador;

- Robustez: capacidade de oferecer predições corretas em presença de dados ruidosos ou dados ausentes;
- Escalabilidade: capacidade de construção de classificadores utilizando-se grandes volumes de dados;
- Interpretabilidade: relaciona-se ao nível de compreensão das predições geradas pelo classificador.

3.4.1 Medidas para Avaliação

A tabela 1 representa uma importante ferramenta para a avaliação de classificadores, chamada matriz de confusão. Nesta tabela, algumas medidas básicas estão apresentadas no contexto da regulação de planos de saúde:

Tabela 1 – Medidas básicas para avaliação de performance

		Predição	
		Não Autorizado	Autorizado
Verdadeiro	Não Autorizado	VP	FN
	Autorizado	FP	VN

Quando o problema de classificação envolve duas classes (classificação binária), como é o caso da regulação de planos de saúde abordada neste trabalho, comummente refere-se a uma das classes como positiva e a outra como negativa. Aqui, a classe "autorizado" representa a classe negativa enquanto que a classe "não autorizado" representa a classe positiva. Dito isso, as medidas VP, FN, FP e VN possuem a seguinte interpretação:

- Verdadeiros Positivos (VP): referem-se aos exemplares que pertencem a classe positiva ("não autorizado") e que foram preditos corretamente como pertencentes a esta classe;
- Falsos Negativos (FN): referem-se aos exemplares que pertencem a classe positiva e foram preditos erroneamente como sendo da classe negativa ("autorizado");
- Falsos Positivos (FP): referem-se aos exemplares pertencentes a classe negativa e que foram preditos incorretamente como da classe positiva;
- Verdadeiros Negativos (VN): referem-se aos exemplares que pertencem a classe negativa e foram preditos corretamente como sendo desta classe.

Diversas outras métricas podem ser calculadas a partir dessas medidas básicas listadas acima. Algumas delas, como a Taxa de Verdadeiros Positivos, Taxa de Falsos

Positivos, Acurácia e Área sob a Curva ROC estão comentadas a seguir. Uma discussão mais completa sobre essas e outras medidas pode ser obtida em Verbiest, Vermeulen e Teredesai (2014) e Han, Kamber e Pei (2011).

A Taxa de Verdadeiros Positivos (TVP, também conhecida como *recall* ou sensibilidade) é calculada pela Equação 3.1 e, segundo a Tabela 1, pode ser entendida como a razão da quantidade de serviços corretamente classificados como "não autorizados" pela soma de todos os serviços pertencentes a classe "não autorizados".

$$\text{TVP} = \frac{\text{VP}}{\text{VP} + \text{FN}} \tag{3.1}$$

A Taxa de Falsos Positivos (TFP) é calculada pela Equação 3.2 e, segundo a Tabela 1, pode ser entendida como a razão da quantidade de serviços incorretamente classificados como "não autorizados" pela quantidade de serviços "autorizados".

$$\text{TFP} = \frac{\text{FP}}{\text{FP} + \text{VN}} \tag{3.2}$$

A Acurácia (AC), uma das medidas mais conhecidas, representa a taxa de exemplares corretamente classificados pela soma de todos os exemplares. Segundo a Tabela 1, a Acurácia pode ser entendida como a razão entre a quantidade de serviços classificados corretamente pela soma da quantidade total de serviços, como mostrado na Equação 3.3. Essa medida é mais indicada nos casos em que a distribuição entre as classes é relativamente balanceada (VERBIEST; VERMEULEN; TEREDESAI, 2014). Nos casos em que há desbalanceamento de classes a Acurácia não é a medida mais adequada, uma vez que ode levar conclusões incorretas (LóPEZ; FERNáNDEZ; HERRERA, 2014).

$$\text{AC} = \frac{\text{VP} + \text{VN}}{\text{VP} + \text{FP} + \text{FN} + \text{VN}} \tag{3.3}$$

A Área sob a Curva ROC ou AUC (do inglês *Area Under Curve*) é uma outra métrica que também serve para medir a acurácia de um classificador e comummente é utilizada em domínios em que há desbalanceamento de classes (HAN; KAMBER; PEI, 2011; LóPEZ; FERNáNDEZ; HERRERA, 2014). Seu cálculo é feito com base na curva ROC, que por sua vez representa a relação entre as métricas Taxa de Verdadeiros Positivos e Taxa de Falsos Positivos. O gráfico que representa a curva ROC bem como a métrica AUC são ferramentas importantes na avaliação de performance de classificadores, principalmente devido a independência da distribuição das classes (VERBIEST; VERMEULEN; TEREDESAI, 2014; SEIFFERT et al., 2014). Uma discussão mais aprofundada sobre AUC e também orientações sobre a utilização dessa métrica em pesquisas podem ser obtidas em Fawcett (2006).

3.4.2 Métodos de Validação

Os métodos de avaliação definem como os dados disponíveis devem ser empregados nas tarefas de construção e avaliação da performance dos classificadores. A seguir são feitas algumas considerações sobre 3 desses métodos, com base nos estudos de Verbiest, Vermeulen e Teredesai (2014) e López, Fernández e Herrera (2014).

O método mais antigo consiste na utilização de todos os dados disponíveis tanto para o treinamento quanto para os testes. Os resultados obtidos durante a avaliação tendem a ser muito otimistas, uma vez que os classificadores já terão conhecimento dos exemplares que serão testados.

Outro método de avaliação conhecido como *houldout*, consiste na divisão dos dados disponíveis em dois conjuntos: um para ser utilizado no treinamento e outro nos testes para a medição da performance. Esse método tem como vantagens a simplicidade e facilidade de uso. Entre as desvantagens, destaca-se o fato de a avaliação ser feita em apenas uma pequena porção dos dados, podendo acontecer de esses dados serem mais fáceis ou mais difíceis de classificar do que aqueles utilizados durante o treinamento, afetando as medidas de performance.

Um dos métodos que procura lidar com as limitações do *houldout* é o *K-fold Cross Validation*. Neste método, os dados disponíveis são divididos em k partes iguais e cada uma dessas partes é avaliada por um classificador construído com base nas k-*1* partes restantes e ao final, as medidas de performance são calculadas pela média das performances obtidas na avaliação de cada parte. Uma das principais vantagens desse método há uma melhor exploração dos dados durante os testes, sendo que cada exemplar é testado uma única vez. A escolha pelo valor de k está relacionada com o viés e variância do classificador a ser obtido: valores pequenos para k, k=*2* por exemplo, tornam a performance do classificador muito dependente da parte que será utilizada nos testes, enquanto que valores maiores de k permitem que todos os conjuntos de dados utilizados no treinamento sejam mais semelhantes entre sí, tornando a performance mais uniforme. Tipicamente são utilizados os valores 5 ou 10.

3.5 Exemplos de Algoritmos de Classificação

A seguir estão descritos alguns dos principais algoritmos de classificação utilizados na literatura e que também foram empregados no neste trabalho nas medições sobre perda e recuperação de performance em bases de dados desbalanceadas.

3.5.1 C4.5

O algoritmo C4.5 (QUINLAN, 1993) constrói árvores de decisão. Durante a construção da árvore, utiliza-se da métrica razão de ganho de informação para selecionar o atributo em que acontecerá a divisão (ALBISUA et al., 2012). Uma vez selecionado o atributo, o conjunto de dados de treinamento é dividido com base no valor desse atributo e o processo de seleção se repete recursivamente. A construção da árvore termina quando houver apenas exemplares de uma dada classe no subconjunto que resulta da divisão ou até que os dados do subconjunto não sejam suficientes para tentar outra divisão (PERNER; APTE, 2004).

Este algoritmo é capaz de lidar com atributos numéricos, nominais (ou categóricos) e também com valores não informados, além de possuir um mecanismo de poda que simplifica a árvore de decisão construída (LEE; LIU; JIN, 2014). É um dos algoritmos mais usados, principalmente como classificador de base em sistemas com múltiplos classificadores e também como parte de técnicas inteligentes de *resampling* aplicadas em problemas que envolvem desbalanceamento de classes (ALBISUA et al., 2012).

3.5.2 RIPPER

O algoritmo *Repeated Incremental Pruning to Produce Error Reduction* (RIPPER) foi proposto por Cohen (1995) e é do tipo que produz classificadores baseados em regras. Este algoritmo produz um conjunto de regras, uma de cada vez, seguindo duas etapas: crescimento e poda.

Na etapa de crescimento, as regras encontram-se inicialmente vazias (sem nenhuma condição) e as condições são adicionadas com o intuito de maximizar a cobertura dos dados de treinamento. Então prossegue-se com a etapa de poda, que procura eliminar condições das regras de tal forma que não afete a acurácia da classificação. Estas etapas se repetem até que todas as classes sejam cobertas pelo algoritmo. Ao final, as regras com as melhores acurácias são mantidas (LORENA et al., 2011).

Uma das vantagens dos classificadores construídos por meio deste algoritmo é que estes são facilmente interpretáveis, uma vez que o conhecimento extraído dos dados é representado por regras de fácil entendimento.

3.5.3 *Support Vector Machines*

As *Support Vector Machines* ou simplesmente SVM representam um dos melhores algoritmos utilizados em tarefas de classificação e regressão (FARQUAD; BOSE, 2012). Possui fundação teórica sólida, pode ser treinado com poucos exemplares e é insensível quanto ao número de dimensões (quantidade de atributos) (WU et al., 2007).

Em um problema de classificação binária, o objetivo do SVM é encontrar a melhor função de classificação capaz de distinguir os membros de duas classes. Em um conjunto de dados separavel linearmente, por exemplo, essa função corresponde a um hiperplano *f(x)* que divide os exemplares em duas classes (WU et al., 2007). Devido ao fato de haver muitos desses hiperplanos, o SVM garante que o melhor deles seja encontrado escolhendo-se aqueles que proporcionem a maior margem entre as duas classes.

O SVM também pode ser empregado em tarefas de classificação envolvendo mais de duas classes, seja por meio da combinação de múltiplos classificadores binários ou por meio de extensões do algoritmo (WANG; LIN, 2014).

3.5.4 *Naive Bayes*

O algoritmo de classificação *Naive Bayes* baseia-se no Teorema de Bayes e é particularmente util nos casos em que a dimensionalidade dos dados é alta, podendo alcançar performance de predição semelhante a de métodos mais sofisticados como os baseados em árvores de decisão e alguns tipos de redes neurais (DENG et al., 2014).

Diferentemente de outros métodos bayesianos, o *Naive Bayes* assume que todos os atributos são condicionalmente independentes. Dessa forma deixa de ser necessário verificar as relações de dependência condicional entre os atributos e a performance do algoritmo pode escalar de forma linear com os dados de treinamento (DENG et al., 2014).

Essa suposição de independência condicional entre os atributos raramente é encontrada em dados reais e esse pode ser um fator capaz de prejudicar a performance de classificação (DENG et al., 2014; WU et al., 2015). No entanto, mesmo nos casos em que essa suposição é violada há evidencias teóricas e experimentais de que é possível obter classificadores de qualidade baseados nesse algoritmo (WU et al., 2015).

3.5.5 *Random Forest*

O *Random Forest* é um algoritmo que produz uma coleção de árvores de decisão não podadas proposto por Breiman (2001). Cada árvore é treinada em um subconjunto dos dados de treinamento (*bootstrap* com reposição) e com um subconjunto dos atributos selecionados aleatoriamente (BROWN; MUES, 2012). Cada novo exemplar que precisa ser classificado é submetido a cada uma dessas árvores, de tal forma que o resultado final da classificação é decidido por meio de votação (BROWN; MUES, 2012). O treinamento das árvores em um subconjunto dos dados e a seleção aleatória de atributos são dois fatores que contribuem para a diversidade dos modelos baseados nesse algoritmo (DITTMAN; KHOSHGOFTAAR; NAPOLITANO, 2015).

Há diversas vantagens na utilização desse algoritmo em problemas de classificação. Dittman, Khoshgoftaar e Napolitano (2015) comentam sobre a robustez em presença

outliers e dados ruidosos, além da simplicidade do uso. Li et al. (2014) também comentam sobre a simplicidade e sobre o fato de a diversidade dos modelos gerados influenciar de forma positiva na performance, além da possibilidade de implementação paralela para reduzir os tempos de execução. Farquad e Bose (2012) destacam que esse algoritmo executa de forma eficiente em grandes bases de dados, que pode ser utilizado para estimar a importância de atributos para a classificação e também para estudar a interação entre atributos.

3.6 Seleção de Atributos

É a tarefa de buscar pelo menor subconjunto de atributos ou mesmo um subconjunto com exatamente k atributos que proporcione erros mínimos de generalização (VERGARA; ESTéVEZ, 2013). Outros benefícios que também podem ser alcançados são a redução de recursos computacionais necessários ao treinamento dos modelos e melhor entendimento dos processos que originam os dados utilizados no treinamento (VERGARA; ESTéVEZ, 2013; GUYON; ELISSEEFF, 2003).

Bolón-Canedo, no e Alonso-Betanzos (2012) comentam sobre diversas formas de classificar os métodos de seleção de atributos. No que diz respeito à relação com os algoritmos aprendizagem, pode-se dividir em 3 tipos: Filtros, *Wrappers* e Embutidos (do inglês *Embedded*). Os métodos do tipo Filtro baseiam-se em medidas extraídas dos próprios dados, não possuem relação com o algoritmo de aprendizagem e geralmente são empregados durante as etapas de pré-processamento (antes do treinamento dos modelos). Os métodos do tipo *Wrapper* utilizam um algoritmo de predição na seleção do melhor subconjunto de atributos. Já os métodos do tipo Embutido ou *Embedded* realizam a seleção de atributos durante o processo de treinamento e geralmente são específicos de um dado algoritmo de aprendizagem.

Os métodos do tipo Filtro ainda podem ser divididos em univariados e multivariados. Os métodos univariados não levam em consideração as relações de dependência entre os atributos, são de execução rápida e escaláveis. Por outro lado, os multivariados consideram essa relação de dependência ao custo de serem de execução mais lenta e também menos escaláveis.

A seguir comenta-se sobre dois métodos de seleção de atributos empregados neste trabalho, ambos do tipo Filtro: Razão de Ganho de Informação e Mínima Redundância Máxima Relevância (do inglês *minimum Redundancy Maximum Relevance* ou mRMR). O primeiro é também do tipo univariado enquanto que o segundo é do tipo multivariado, ou seja, que considera as relações de dependência entre os atributos selecionados.

3.6.1 Razão de Ganho de Informação

O método da Razão de Ganho de Informação (RGI) (QUINLAN, 1993) representa um refinamento de um outro método chamado Ganho de Informação (GI) que por sua vez baseia-se no conceito de entropia oriundo da área de teoria da informação. Basicamente, GI mede o quanto o grau de incerteza da predição reduz em função do valor de um dado atributo X (DESSÌ; PES, 2015):

$$GI(C|X) = H(C) - H(C|X) \tag{3.4}$$

sendo que *H(C)* representam a entropia da classe antes e depois de observado o atributo X.

O resultado da Equação 3.4 favorece os atributos que possuem uma quantidade maior de valores possíveis (ROKACH; MAIMON, 2010a). O método RGI procura atenuar esse ponto negativo dividindo-se o valor do ganho de informação pela entropia do atributo em questão, segundo a Equação 3.5:

$$RGI(C|X) = \frac{GI(C|X)}{H(X)}. \tag{3.5}$$

3.6.2 Mínima Redundância Máxima Relevância

O método Mínima Redundância Máxima Relevância (do inglês *minimum Redundancy Maximum Relevance* ou mRMR) (PENG; LONG; DING, 2005) seleciona os atributos de forma a maximizar a relevância para com o atributo alvo e minimizar a redundância em relação aos demais atributos. Ambos os critérios são calculados com base na medida de informação mútua (BOLóN-CANEDO et al., 2014). O resultado da execução desse método é uma lista de atributos ordenados, cuja interpretação é a seguinte (LI et al., 2012): um dado subconjunto formado pelos primeiros k atributos da lista possui melhor relação relevância versus redundância que o sobconjunto formado pelos primeiros $k+1$ atributos.

Há diversas formas de combinar as medidas de redundância e relevância no método mRMR, sendo que as duas mais utilizadas são a diferença e o quociente. Esses duas formas são representadas pelos parâmetros MID (diferença) e MIQ (quociente) na implementação disponibilizada pelos autores desse método (MRMR, 2016).

3.7 Balanceamento de Classe

A aprendizagem de classificadores utilizando-se bases de dados desbalanceadas representa um importante problema de mineração de dados, caracterizado pela distribuição

desigual entre as quantidades de exemplares de cada classe na base de dados (FERNáNDEZ; GARCÍA; HERRERA, 2011; HE; GARCIA, 2009).

He e Garcia (2009) comentam que o desbalanceamento de classes pode ser ocasionado por fatores inerentes ao domínio dos dados. Para o caso da regulação de planos de saúde, sabe-se que ocorrem muito mais serviços autorizados do que não autorizados. Nesses casos o desbalanceamento é do tipo intrínseco. Há também os casos em que o desbalanceamento de classes é causado por fatores externos como intervalos de tempo ou limitações de armazenamento. Para esses casos o desbalanceamento é chamado extrínseco.

Muitos algoritmos de classificação consideram que sejam utilizadas bases de dados balanceadas durante o treinamento. Isso leva a obtenção de classificadores que atingem uma boa cobertura quanto aos exemplares da classe majoritária enquanto que os aqueles da classe minoritária são frequentemente classificados incorretamente. Entre as razões para esse comportamento, destacam-se (LóPEZ et al., 2013):

- Uso de medidas de performance globais, como a acurácia, pode favorecer a classe majoritária;
- As regras que classificam os exemplares da classe minoritária geralmente são muito especializadas e possuem baixa cobertura e comummente são descartadas pelos algoritmos em favor de regras mais genéricas;
- O impacto na performance dos classificadores causado por exemplares ruidosos da classe minoritária é maior devido a baixa quantidade de exemplares.

López et al. (2013) também comenta que as soluções desenvolvidas para tratar o problema do desbalanceamento de classes podem ser agrupadas em três tipos:

- Baseadas em amostragem dos dados: procuram modificar a quantidade dos exemplares utilizados no treinamento dos algoritmos para produzir uma base de dados melhor balanceada;
- Baseadas em algoritmos: baseiam-se em modificações nos algoritmos de aprendizagem para que estes considerem o desbalanceamento de classes em suas implementações;
- Baseadas em custos: procuram atribuir um custo maior para a classificação incorreta do exemplares da classe minoritária por meio de alterações nos dados usados no treinamento, nos algoritmos ou em ambos.

A seguir há uma breve descrição sobre alguns desses métodos de tratamento.

3.7.1 *Random Undersampling* e *Random Oversampling*

Os métodos baseados em amostragem, também conhecidos como métodos de *resampling*, são capazes de melhorar a performance de predição na maioria das aplicações em que há desbalanceamento de classes (HE; GARCIA, 2009). A principal vantagem desses métodos consiste na independência dos algoritmos de aprendizagem utilizados nessas aplicações (GALAR et al., 2012).

Os métodos de *resampling* podem ser divididos em dois tipos (SOBHANI; VIKTOR; MATWIN, 2014): *undersampling* e *oversampling*. Os métodos do tipo *undersampling* baseiam-se na redução da quantidade de exemplares da classe majoritária, enquanto que os métodos de *oversampling* procuram aumentar a quantidade de exemplares da classe minoritária. Entre os métodos de resampling há dois que são muito comuns: *Random Undersampling* e *Random Oversampling*.

O método *Random Undersampling* procura balancear as classes em uma base de dados por meio da eliminação aleatória dos exemplares da classe majoritária e possui como desvantagem a possibilidade de descartar exemplares que podem ser potencialmente úteis ao aprendizado (GALAR et al., 2012).

O método *Random Oversampling*, de forma análoga ao método anterior, funciona por meio da replicação aleatória dos exemplares da classe minoritária. Diversos autores afirmam que este método pode aumentar a possibilidade de ocorrer *overfitting* (GALAR et al., 2012), situação na qual o classificador construído especializa-se e atinge uma performance de predição alta durante o treinamento e abaixo do esperado durante os testes.

3.7.2 SMOTE

Synthetic Minority Over-sampling Technique, ou simplesmente SMOTE, é um método do tipo *oversampling* que constrói exemplares artificiais da classe minoritária por meio da interpolação entre um dado exemplar da classe minoritária e seus k vizinhos mais próximos (CHAWLA, 2002a).

Apesar de ser um dos métodos mais conhecidos e da capacidade de aumentar a quantidade de instâncias da classe minoritária sem o mesmo risco de *overfiting* inerente ao método *Random Oversampling*, o SMOTE pode proporcionar resultados ruíns quando utilizado de forma isolada (sem extensões). Isso deve-se principalmente a forma como as exemplares artificiais são construídos, levando-se em consideração apenas a distância e ignorando outras características dos dados, como a presença de dados ruidosos ou sobreposição de classes (*overlapping*) (SáEZ et al., 2015). Entre esses pontos negativos, destacam-se os seguintes:

- criação de exemplares em vizinhanças que não contribuem para o aprendizado da classe minoritária;
- introdução de exemplares ruidosos em áreas em que a maioria dos exemplares pertencem a classe majoritária e consequentemente aumentando a sobreposição das classes.

3.7.3 MetaCost

MetaCost é um meta-algoritmo de classificação proposto por Domingos (1999) e que pode ser empregado em conjunto com qualquer algoritmo de classificação (AGGARWAL, 2014b). O seu funcionamento consite na construção de um conjunto de classificadores (*ensemble*) utilizando subconjuntos dos dados de treinamento (*bagging*). Este *ensemble*, cuja saída é baseada em probabilidades, é então utilizado para reclassificar (mudar os rótulos) cada exemplar de treinamento de tal forma que o novo rótulo atribuido minimize o custo esperado de classificação. Em seguida, os rótulos originais são descartados e um novo classificador é construído utilizando-se os exemplares com os rótulos modificados (WITTEN; FRANK; HALL, 2011).

Uma das vantagens do MetaCost, por ser um meta-algoritmo, é poder empregar internamente qualquer algoritmo de classificação que não seja baseado em custos. O desafio dessa abordagem é que a reatribuição dos rótulos representa uma etapa de risco desse método. Este risco está relacionado ao fato de que as probabilidades obtidas pelo *ensemble* podem não refletir as probabilidades intrínsecas ao domínio dos dados (AGGARWAL, 2014b).

4 Metodologia

A investigação dos efeitos do desbalanceamento de classes no aprendizado da regulação de planos de saúde apresentada neste trabalho segue o experimento proposto por Prati, Gustavo E A P e Silva (2014). Em linhas gerais, esse experimento investiga como varia a performance de predição de classificadores antes e depois da utilização de métodos de tratamento para desbalanceamento de classes. No decorrer do experimento são utilizadas diversas bases de dados em que a distribuição das classes foi modificada artificialmente, vários algoritmos de classificação e métodos de tratamento para desbalanceamento de classes.

As seções seguintes apresentam uma visão geral dos dados utilizados no experimento, bem como uma explicação mais detalhada das diversas etapas.

4.1 Definição dos Dados

Os dados utilizados neste estudo compreendem solicitações de serviços médicos e odontológicos pertencentes a duas operadoras de planos de saúde distintas, sendo uma delas com atuação na cidade de Teresina, PI e o outra em Recife, PE.

A operadora que atua em Teresina administra um plano de saúde que possui 50000 beneficiários e os dados fornecidos por ela estão organizados em duas bases distintas. A primeira dessas bases contém apenas solicitações de serviços odontológicos realizadas no período entre abril de 2014 a março de 2015 em decorrência do atendimento prestado a 4114 beneficiários e que será referenciada neste trabalho pelo nome de Teresina Odontológico. A segunda base contém apenas solicitações de serviços ambulatoriais (consultas e exames médicos ambulatoriais) realizados em decorrência ao atendimento prestado a 7080 beneficiários no período entre julho de 2014 a maio de 2016 e que será referenciada neste trabalho de Teresina Ambulatorial.

Já os dados obtidos da operadora que atua em Recife são oriundos de um plano de saúde que atende ao todo 23000 beneficiários e contém apenas solicitações de serviços odontológicos. Essas solicitações foram realizadas em decorrência ao atendimento de 14819 beneficiários no período entre os meses de agosto de 2007 a março de 2016. Neste trabalho, esta base de dados é referenciada pelo nome Recife Odontológico.

As bases de dados Teresina Ambulatorial e Recife Odontológico foram obtidas com todos os atributos e rótulos definidos, prontas para a construção de classificadores. Já a base de dados Teresina Odontológico faz parte de um outro estudo do mesmo grupo de pesquisa deste trabalho, ainda não terminado, sobre construção e seleção de atributos no

domínio da regulação de planos de saúde odontológicos. Uma explicação mais detalhada sobre como seus atributos e rótulos foram definidos encontra-se no Apêndice A.

A base de dados Recife Odontológico possuía originalmente 223150 exemplares e teve essa quantidade reduzida para 15% desse valor, removendo-se aleatoriamente os exemplares e mantendo-se a distribuição de classes original, ficando ao final com 33472 exemplares. Essa redução fez-se necessária para viabilizar a execução dos experimentos com os recursos computacionais disponíveis e encontra-se melhor detalhada na Seção 4.3.2.

Em resumo, os dados utilizados neste trabalho estão sumarizados na Tabela 2.

Tabela 2 – Sumário dos dados utilizados neste trabalho.

	Teresina Odontológico	Teresina Ambulatorial	Recife Odontológico
Beneficiários	4114	7080	14819
Serviços solicitados	28042	17456	223150
Profissionais solicitantes	70	572	83
Profissionais de regulação	8	4	6
Período	abril de 2014 a março de 2015	julho de 2014 a maio de 2016	agosto de 2007 a março de 2016
Quantidade de atributos	84	17	8
Balanceamento de classes	94% autorizados e 6% não autorizados	98% autorizados e 2% não autorizados	94% autorizados e 6% não autorizados

Do ponto de vista do balanceamento de classes, todas as solicitações de serviços estão divididas entre duas classes que representam as solicitações autorizadas e não autorizadas. As discrepâncias entre as quantidades das duas classes caracterizam estas bases de dados como desbalanceadas (STEFANOWSKI, 2013). O desbalanceamento encontrado nessas bases é considerado do tipo intrínseco, ou seja, inerente ao domínio da regulação de planos de saúde, em que comumente ocorrem mais serviços autorizados do que não autorizados (HE; GARCIA, 2009).

4.2 Descrição do Experimento

A execução do experimento pode ser dividida em 4 etapas, que são: divisão das bases de dados, medição da perda de performance, medição da recuperação de performance e análise dos resultados.

Na primeira etapa é feita a divisão das bases de dados em bases menores, destinadas ao treinamento e teste dos classificadores. Primeiro, 25% dos exemplares de cada uma das três bases de dados utilizadas neste trabalho é reservado para a execução de testes dos classificadores, mantendo-se a mesma distribuição de classes encontrada nas bases de dados antes da divisão. Já as bases de dados que serão utilizadas no treinamento possuem as quantidades de exemplares limitadas à quantidade de exemplares da classe minoritária encontrada nos 75% restantes dos dados. Essa limitação permite que seja possível criar diversas bases de dados com variadas distribuições de classes. As distribuições de classes

adotadas são: 1/99, 5/95, 10/90, 20/80, 30/70, 40/60, 50/50 (base de dados balanceada), 60/40, 70/30, 80/20, 90/10, 95/5 e 99/1. Logo, cada uma das 3 bases de dados utilizadas neste trabalho dão origem a 14 bases menores, sendo 13 delas com variadas distribuições de classe e destinadas ao treinamento e 1 com a distribuição de classes original destinada aos testes dos classificadores.

Feita a divisão das bases de dados, segue-se para a segunda etapa em que é feito o treinamento e teste de diversos algoritmos de classificação com o intuito de medir a relação entre a distribuição de classes e a performance de predição obtida. Sobre os algoritmos de classificação, procurou-se escolher alguns que fossem representativos dos diversos paradigmas e cujas implementações estivessem disponíveis na ferramenta *WEKA* (HALL et al., 2009), versão 3.7.13. São eles: C4.5, Ripper, *Random Forest*, SVM e *Naïve Bayes*. Esses algoritmos foram utilizados programaticamente e não via interface gráfica, seguindo as orientações disponíveis em (USE. . . , 2016).

A Área sob a Curva ROC (também conhecida como *Area Under Curve* ou AUC) é a métrica utilizada nestre trabalho para medir a performance de predição. Essa métrica é a mesma utilizada no estudo que embasou este experimento e é considerada apropriada para a avaliação de performance em tarefas de classificação envolvendo bases de dados desbalanceadas por diversos outros autores (ALBISUA et al., 2012; LIU; WU; ZHOU, 2009).

Após os testes dos classificadores é feita a medição que avalia o quanto da performance dos classificadores foi perdida devido ao desbalanceamento de classes. Para isso, a performance de um classificador treinado em uma base de dados cuja distribuição de classes é representada por I é então comparada com um valor de referência segundo a Equação 4.1:

$$L = (B - I)/B \tag{4.1}$$

Na equação 4.1, L (do inglês *Loss*) representa o valor da perda de performance, B (do inglês *Balanced*) representa o valor de performance na distribuição de classes 50/50 (tomada como valor de referência), I (do inglês *Imbalanced*) representa o valor da performance de um classificador que foi treinado em uma base de dados com uma das distribuições de classes estudadas. Por meio dos dados calculados pela Equação 4.1 é possível observar como a performance de cada algoritmo de classificação varia segundo as diversas distribuições de classe estudadas, bem como qual algoritmo é mais ou menos afetado pelo desbalanceamento de classes.

Terminado o cálculo dos valores da Equação 4.1, o experimento segue para a terceira etapa que envolve o tratamento dos dados utilizados e em seguida é realizado novamente o treinamento e teste dos classificadores. O objetivo é verificar como a performance dos

classificadores varia após a aplicação dos métodos de tratamento aos dados utilizados no treinamento, observando o quanto é possível recuperar da performance que foi perdida devido ao desbalanceamento de classes. Os métodos de tratamento empregados são: SMOTE (CHAWLA, 2002b), MetaCost (DOMINGOS, 1999) e *Random Oversampling* (ALBISUA et al., 2012; BATISTA; PRATI; MONARD, 2004).

A escolha por estes métodos deve-se ao fato de estarem entre aqueles utilizados no estudo que embasou este experimento e também pela disponibilidade de implementação utilizando-se de componentes pré-existentes na ferramenta *WEKA*. Nenhum destes métodos encontra-se disponível por padrão na ferramenta *WEKA*, versão 3.7.13: os métodos SMOTE e MetaCost foram instalados como *plugins* (ambos com versão 1.0.3). Já o método *Random Oversampling* foi desenvolvido com base na *api* de programação disponibilizada por essa ferramenta. De forma análoga ao uso dos algoritmos de classificação, esses métodos de tratamento também foram utilizados de forma programática e não via interface gráfica, seguindo-se as orientações disponíveis em (USE. . . , 2016).

Após a construção dos classificadores com os dados tratados, a recuperação da performance é avaliada da seguinte forma: primeiro, mede-se a perda de performance atingida pelos classificadores com os dados tratados utilizando a Equação 4.2; segundo, combina-se os resultados de perda de performance antes e depois do tratamento das bases de dados por meio da Equação 4.3.

$$L_t = (B - T)/B \tag{4.2}$$

$$R = (L - L_t)/L \tag{4.3}$$

Na equação 4.2, L_t representa a medida de perda de performance após testar um classificador treinado em uma base de dados tratada. B representa a performance de referência, obtida pelo classificador treinado com a base de dados balanceada (50/50) e não tratada por nenhum método. T (do inglês *Treated*) representa a performance obtida pelo classificador treinado em uma base de dados desbalanceada e tratada por um dos métodos de tratamento.

Com os resultados da Equação 4.2 é possível fazer diversas observações. Por exemplo, quais métodos de tratamento melhor beneficiam cada algoritmo de classificação. Também é possível observar como varia as perdas de performance por distribuição de classes para cada método de tratamento e com isso concluir que, para uma dada distribuição de classes, um determinado método de tratamento pode levar a maiores ou menores perdas de performance que os demais.

Na equação 4.3, R (do inglês *Recovery*) representa a medida de recuperação de performance. L é a medida de perda de performance nas bases de dados desbalanceadas

(equação 4.1) e L_t é a medida de perda de performance nas bases tratadas (Equação 4.2). O resultado pode ser compreendido da seguinte forma: $R = 100\%$ quando $L_t = 0\%$, ou seja, a recuperação é de 100% quando a perda de performance de um dado algoritimo na base de dados tratada é de 0% (performance igual à da base de dados não tratada e balanceada). Por outro lado, R = 0% quando $L_t = L$, ou seja, quando a perda de performance utilizando-se uma base de dados tratada for igual a perda de performance utilizando-se uma base não tratada.

Por meio dos resultados da Equação 4.3 é possível perceber como ocorre a recuperação de performance para cada combinação de algoritmo de classificação, método de tratamento e distribuição de classes. Mais especificamente, é possível observar em que distribuição de classes a recuperação é maior e também quais algoritmos se beneficiam mais ou menos de cada método de tratamento.

Neste experimento, todo o processo que vai desde a divisão das bases de dados até o treinamento e teste dos algoritmos de classificação nos dados tratados é repetido por 100 vezes. Essa repetição permite que os resultados das Equações 4.1, 4.2 possam ser analisadas segundo valores médios e intervalos de confiança, aumentando as possibilidades de análise dos resultados. O valor 100 foi o mesmo utilizado no estudo que serviu de embasamento para este trabalho e acredita-se, empiricamente, que seja um valor apropriado para a obtenção dos intervalos de confiança.

Por fim, na ultima etapa que é a de análise dos resultados, as medidas de perda e recuperação de performance são resumidas em valores médios e intervalos de confiança e também agrupadas por distribuição de classe, algoritmo de classificação e método de tratamento.

4.3 Limitações

As principais limitações deste trabalho são: quantidade de bases de dados disponíveis, quantidade de exemplares, distribuição de classes e escopo das análises.

4.3.1 Quantidades de Bases de Dados Disponíveis

Em seu estudo, (PRATI; Gustavo E A P; SILVA, 2014) empregou 22 bases de dados de domínio diferentes, a maioria delas disponível publicamente. Já este trabalho emprega apenas 3, todas sobre regulação de planos de saúde e cujos dados são de acesso restrito. Este contexto naturalmente dificulta a realização desse estudo utilizando uma quantidade maior de bases de dados. Acredita-se que os resultados que podem ser encontrados utilizando-se essas 3 bases de dados são representativos para o domínio da regulação de planos de saúde.

4.3.2 Quantidade de Exemplares

A quantidade de exemplares pode influenciar a execução do experimento apresentado neste trabalho de duas maneiras:

- quando é pequena demais, ao ponto de a quantidade de instâncias da classe minoritária ser insuficiente para a aprendizagem;
- quando é grande demais, ao ponto de tornar a execução do experimento inviável com os recursos computacionais disponíveis.

Se a quantidade total de exemplares for pequena, então a quantidade de instâncias da classe minoritária será menor ainda devido ao desbalanceamento de classes. Logo, aquelas bases de dados utilizadas no treinamento dos classificadores que possuem as distribuições mais acentuadas (1/99, 99/1) podem assumir uma quantidade de exemplares da classe minoritária tão limitada ao ponto de ser insuficiente para a aprendizagem do classificador.

Por outro lado, quando a quantidade de exemplares é muito grande, o tempo necessário para a execução do experimento pode ser inviável. Por exemplo, na etapa do experimento em que os métodos de tratamento são aplicados, cada base de dados que foi dividida por 100 vezes em 13 bases de dados menores é tratada por 3 métodos de tratamento diferentes e em seguida são construídos classificadores utilizando-se 5 algoritmos, totalizando 19500 (100 * 13 * 3 * 5) tarefas de tratamento de dados, treinamento e teste de classificadores.

Entre essas tarefas, algumas são mais computacionalmente custosas do que outras, por exemplo: as tarefas que envolvem o tratamento de dados pelo método *Random Oversampling* e a construção de classificador segundo o algoritmo *Naive Bayes* possuem um tempo de execução muito menor do que as tarefas que envolvem os métodos de tratamento SMOTE e MetaCost e os algoritmos *Random Forest* e SVM. Em especial, o método de tratamento MetaCost e o algoritmo de classificação *Random Forest* empregam internamente *ensembles* ou combinação de classificadores, o que torna suas utilizações mais custosas do ponto de vista computacional.

Foi justamente a grande quantidade de exemplares que motivou a redução da base de dados Recife Odontológico para 15% da quantidade original. Em tentativas de executar a etapa de medição da recuperação de performance com a quantidade total de exemplares, a execução de 6747 tarefas [1] demoram cerca de 1120 minutos (pouco mais de 18 horas) em uma máquina virtual do tipo *c4.8xlarge* com 36 processadores virtuais (AMAZON...,

[1] Neste ponto, a palavra "tarefa" denota a computação que compreende o tratamentos de uma base de dados mais o treinamento e teste de um classificador.

2016). Considerando que o tempo necessário para executar o restante das tarefas mantenha a mesma proporção, a execução de todas as tarefas poderia levar cerca de 54 horas.

4.3.3 Distribuição de Classes

A distribuição de classes afeta a quantidade de instâncias nas bases de dados utilizadas no treinamento, pois a quantidade de exemplares nessas bases de dados deve ser igual a quantidade de exemplares da classe minoritária encontrados nos 75% da base de dados original, que resta após reservar-se 25% do total de exemplares para compor a base de testes.

Durante a realização deste trabalho ocorreu a possibilidade de usar uma quarta base de dados contendo solicitações de serviço relacionadas a atendimentos de urgência. Essa base de dados possui 4086 exemplares no total, sendo 3984 solicitações autorizadas e apenas 102 não autorizada. Logo percebe-se a pequena quantidade de exemplares e o desbalanceamento de classes elevado quando comparada com as outras 3 bases de dados empregadas neste estudo. Após reservar 25% desses dados para a base de testes, os 75% restantes continham apenas 77 exemplares da classe minoritária. Como consequência, as bases de dados com a distribuição de classes modificadas foram limitadas ao tamanho de 77 exemplares, sendo que para as bases em que a distribuição é 1/99 ou 99/1, a quantidade de exemplares da classe minoritária seriam de apenas 1 exemplar. Nessas condições é razoável supor que a aprendizagem dos classificadores estaria seriamente afetada além de ser impossível a utilização do método SMOTE. Por conta desses problemas essa base de dados foi descartada.

4.3.4 Escopo das Análises

Neste trabalho, de forma análoga ao trabalho de Prati, Gustavo E A P e Silva (2014), o desbalanceamento de classes é investigado em sentido amplo, considerando diversas bases de dados diferentes. No entanto há uma diferença: todas as bases de dados utilizadas neste trabalho pertencem a um único domínio que é a regulação de planos de saúde, ao contrário do outro trabalho que utilizou-se de bases de dados de diversos domínios diferentes. Por conta disso não foram feitas análises específicas considerando-se cada uma das bases de dados separadamente, por exemplo: não foi apresentado nenhum resultado sobre como o desbalanceamento de classes da base de dados Teresina Odontológico afeta a performance de predição dos classificadores ou mesmo como o método de tratamento SMOTE comporta-se quando aplicado à base de dados Recife Odontológico.

Uma análise específica para cada base de dados pode trazer informações relevantes. Por exemplo: pode-se descobrir outras distribuições de classe diferentes de 50/50 que resultem na obtenção de classificadores com maiores performances de predição (WEISS;

PROVOST, 2003).

5 Resultados e Discussões

Neste capítulo os resultados das medições sobre perda e recuperação de performance são apresentados e discutidos. Para o melhor entendimento dos resultados, deve-se salientar que as bases de dados em que as distribuições de classes são 1/99, 5/95, 10/90, 20/80, 30/70 e 40/60, a classe minoritária é aquela que representa as solicitações de serviços autorizadas na regulação. De forma análoga, as outras bases em que os níveis de balanceamento variam de 60/40 até 99/1 são aquelas em que a classe minoritária representa as solicitações que foram não autorizadas.

5.1 Perda de Performance

As performances dos classificadores obtidos utilizando-se os dados não tratados (originais) estão sumarizadas na Tabela 3 e Figura 3, que mostram como a performance de cada um dos algoritmos estudados variou segundo as diversas distribuições de classes. De forma complementar, a Tabela 4 e a Figura 4 apresentam os valores médios da perda de performance calculados segundo a Equação 4.1.

Tabela 3 – Valores médios de performance medidos pela Área sob a Curva ROC (AUC)

	1/99	5/95	10/90	20/80	30/70	40/60	50/50	60/40	70/30	80/20	90/10	95/5	99/1
C4.5	0.50	0.50	0.50	0.52	0.59	0.65	0.66	0.65	0.58	0.53	0.50	0.50	0.50
RIPPER	0.50	0.50	0.51	0.53	0.56	0.60	0.61	0.60	0.57	0.53	0.51	0.50	0.50
Random Forest	0.62	0.67	0.70	**0.72**	**0.72**	**0.72**	**0.72**	0.71	0.71	0.69	0.66	0.62	0.54
SVM	0.50	0.50	0.50	0.52	0.54	0.56	0.61	0.59	0.55	0.52	0.51	0.50	0.50
Naive Bayes	0.56	0.62	0.64	0.66	0.66	0.66	0.66	0.66	0.65	0.64	0.62	0.59	0.55
Média	0.54	0.56	0.57	0.59	0.61	0.64	0.65	0.64	0.61	0.58	0.56	0.54	0.52

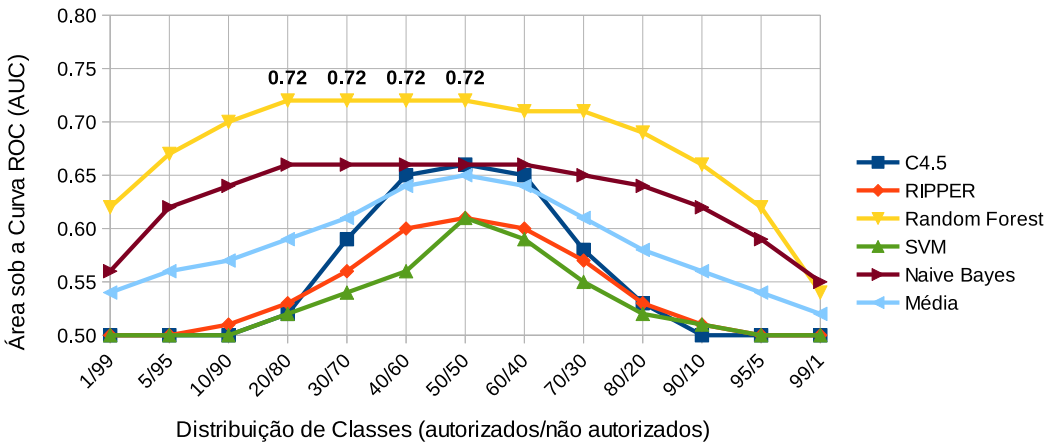

Figura 3 – Valores médios de performance medidos pela Área sob a Curva ROC (AUC).

Tabela 4 – Valores médios de perda de performance calculados segundo a Equação 4.1.

	1/99	5/95	10/90	20/80	30/70	40/60	50/50	60/40	70/30	80/20	90/10	95/5	99/1
C4.5	24%	24%	24%	21%	11%	2%	0%	2%	12%	21%	24%	24%	24%
RIPPER	18%	18%	17%	14%	8%	3%	0%	2%	7%	13%	17%	18%	18%
Random Forest	14%	6%	3%	1%	0%	-1%	0%	1%	2%	4%	9%	14%	25%
SVM	18%	18%	18%	16%	13%	8%	0%	4%	11%	15%	17%	18%	18%
Naive Bayes	16%	6%	2%	0%	0%	0%	0%	0%	1%	3%	6%	10%	16%
Média	18%	15%	13%	10%	6%	2%	0%	2%	7%	11%	15%	17%	20%

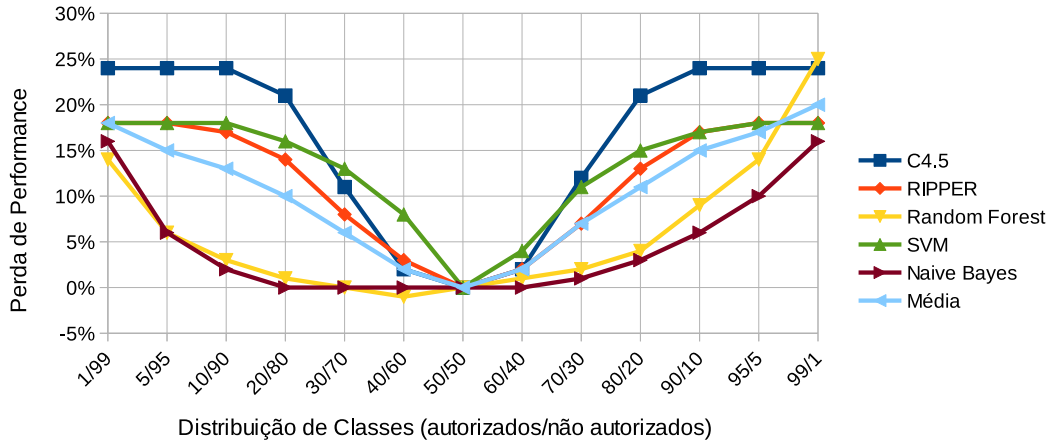

Figura 4 – Valores médios de perda de performance calculados segundo a Equação 4.1.

Observa-se pela Figura 3 que os algoritmos de classificação estudados são afetados pelo desbalanceamento de classes de forma distinta. Os algoritmos *Random Forest* e *Naive Bayes* são os que apresentam as menores variações de performance, inclusive mantendo valores constantes entre as distribuições de classe 20/80 e 50/50. Já os algoritmos C4.5, RIPPER e SVM tiveram as maiores variações nos valores de AUC entre as distribuições de classe 10/90 e 90/10. Nas distribuições de classe mais extremas, em que a classe minoritária representa 10% ou menos do total de exemplares, praticamente não houve variação de performance.

Os resultados com relação ao algoritmo SVM diferem daqueles observados em Prati, Gustavo E A P e Silva (2014), nos quais o SVM foi pouco afetado pelo desbalanceamento de classes. Acredita-se que os parâmetros escolhidos por padrão para a execução deste algoritmo possam ter influenciado esta diferença nos resultados.

Essas mesmas conclusões podem ser obtidas de outra forma, por meio da Figura 4, em que percebe-se os algoritmos *Random Forest* e *Naive Bayes* com valores próximos de 0 entre as distribuições de classe 20/80 e 80/20, ao passo que os demais algoritmos apresentam valores elevados de perda de performance nesse mesmo intervalo, estabilizando-se quando a classe minoritária representa 10% ou menos do total de exemplares.

5.2 Recuperação de Performance

Os valores médios de recuperação de performance, calculados segundo a Equação 4.3, encontram-se sumarizados nas tabelas 5, 6 e 7. Nessas tabelas, os espaços vazios devem-se ao fato de que apenas os valores de perda de performance maiores que 10% terem sido levados em consideração durante o cálculo da recuperação de performance. Essa interpretação segue a orientação do estudo que serviu de embasamento para este trabalho e permite que a análise se concentre nas distribuições de classe em que ocorrem as maiores perdas de performance, evitando-se ruídos causados por valores baixos no denominador da Equação 4.3.

Tabela 5 – Valores médios de recuperação de performance obtidos pelo método *Random Oversampling* segundo a Equação 4.3.

	1/99	5/95	10/90	20/80	30/70	40/60	50/50	60/40	70/30	80/20	90/10	95/5	99/1
C4.5	10%	31%	48%	64%	72%	64%	-	94%	81%	71%	50%	28%	7%
RIPPER	9%	39%	68%	99%	88%	66%	-	67%	87%	99%	75%	42%	10%
Random Forest	9%	8%	20%	-	-	-	-	-	-35%	-7%	-7%	-6%	-4%
SVM	5%	18%	32%	44%	39%	-4%	-	72%	77%	64%	53%	37%	10%
Naive Bayes	-7%	-27%	-46%	-	-	-	-	-	-4%	29%	32%	32%	-3%

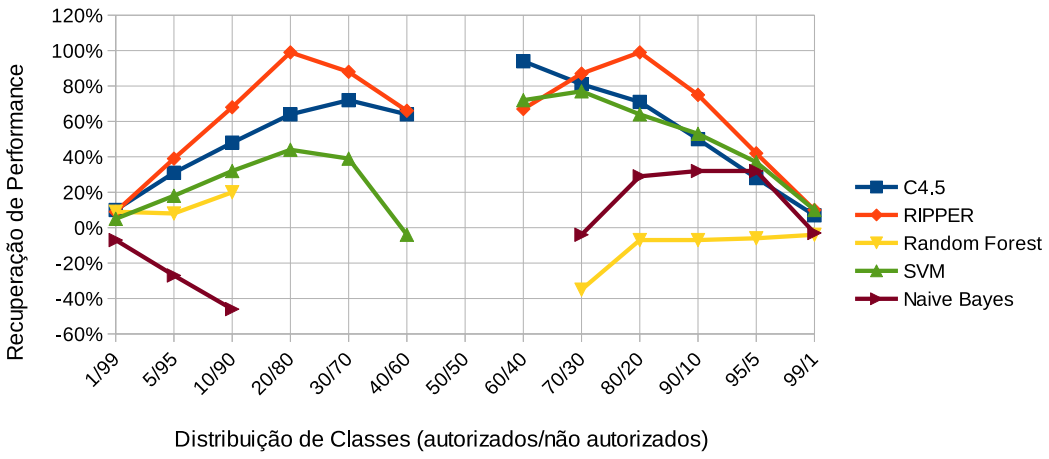

Figura 5 – Valores médios de recuperação de performance obtidos pelo método *Random Oversampling* segundo a Equação 4.3.

A Tabela 5 juntamente com a Figura 5 mostram os valores médios de recuperação de performance obtidos ao se aplicar o método de tratamento *Random Oversampling* aos dados utilizados no treinamento dos classificadores. É possível notar que, para este método de tratamento, os algoritmos que mais foram afetados pelo desbalanceamento de classes são também aqueles que mais se beneficiaram, como é o caso dos algoritmos C4.5, RIPPER e SVM. Os algoritmos *Random Forest* e *Naive Bayes* apresentaram baixos valores de perda de performance, como pode ser visto pela Figura 4 e por isso apresentam poucos pontos no gráfico mostrado na Figura 5. Mesmo assim, nota-se que os valores registrados são

próximos de zero ou negativos, indicando que a performance medida pela métrica AUC após aplicar o método *Random Oversampling* foi igual ou inferior à obtida utilizando-se os dados originais (não tratados por nenhum método).

Tabela 6 – Valores médios de recuperação de performance obtidos pelo método SMOTE segundo a Equação 4.3.

	1/99	5/95	10/90	20/80	30/70	40/60	50/50	60/40	70/30	80/20	90/10	95/5	99/1
C4.5	13%	37%	57%	79%	91%	65%	-	8%	1%	1%	-1%	0%	0%
RIPPER	7%	28%	38%	76%	73%	74%	-	18%	-1%	4%	3%	0%	0%
Random Forest	18%	13%	34%	-	-	-	-	-	-35%	-11%	-7%	-14%	-8%
SVM	23%	42%	48%	57%	58%	83%	-	-15%	-19%	-9%	-3%	-1%	0%
Naive Bayes	-20%	-47%	-94%	-	-	-	-	-	-36%	-35%	-28%	-23%	-4%

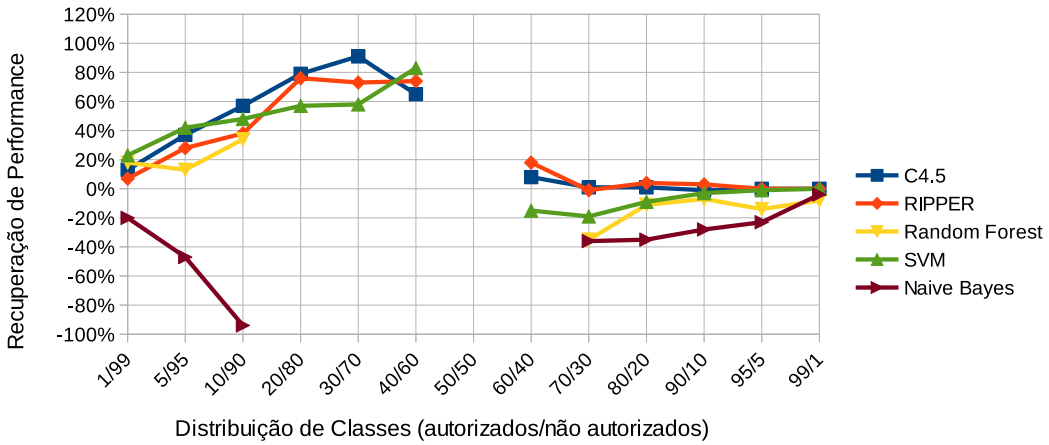

Figura 6 – Valores médios de recuperação de performance obtidos pelo método SMOTE segundo a Equação 4.3.

A Tabela 6 e a Figura 6 mostram os resultados sobre a medição de recuperação de performance obtidos após a utilização do método SMOTE e uma característica chama a atenção: que é a aparência assimétrica do gráfico, indicando que a recuperação de performance foi maior quando a classe minoritária era aquela que representa os serviços autorizados na regulação. Neste caso, o comportamento foi, de certa forma, semelhante àquele proporcionado pelo método *Random Oversampling*, em que os algoritmos C4.5, RIPPER e SVM obtiveram os maiores valores de recuperação de performance enquanto que os algoritmos *Random Forest* e *Naive Bayes* obtiveram valores menores ou mesmo não foram computados devido às medidas de perda de performance abaixo do valor de 10%. No outro caso, em que a classe minoritária representa os serviços não autorizados na regulação, os valores de recuperação de performance foram bem menores, próximos de zero ou mesmo negativos. Este resultado pode estar relacionado com as características negativas do método SMOTE comentadas por Sáez et al. (2015): ao criar instâncias artificiais levando-se em consideração apenas medidas de distância pode-se estar também aumentando a quantidade de instâncias ruidosas ou mesmo a sobreposição entre as classes.

Tabela 7 – Valores médios de recuperação de performance obtidos pelo método MetaCost segundo a Equação 4.3.

	1/99	5/95	10/90	20/80	30/70	40/60	50/50	60/40	70/30	80/20	90/10	95/5	99/1
C4.5	4%	16%	35%	70%	82%	53%	-	80%	85%	69%	41%	15%	3%
RIPPER	7%	32%	61%	79%	67%	47%	-	69%	77%	82%	64%	35%	8%
Random Forest	15%	7%	10%	-	-	-	-	-	14%	20%	-2%	5%	10%
SVM	1%	3%	6%	9%	6%	-7%	-	7%	17%	10%	5%	3%	1%
Naive Bayes	-4%	-17%	-53%	-	-	-	-	-	-50%	-61%	-28%	-26%	-19%

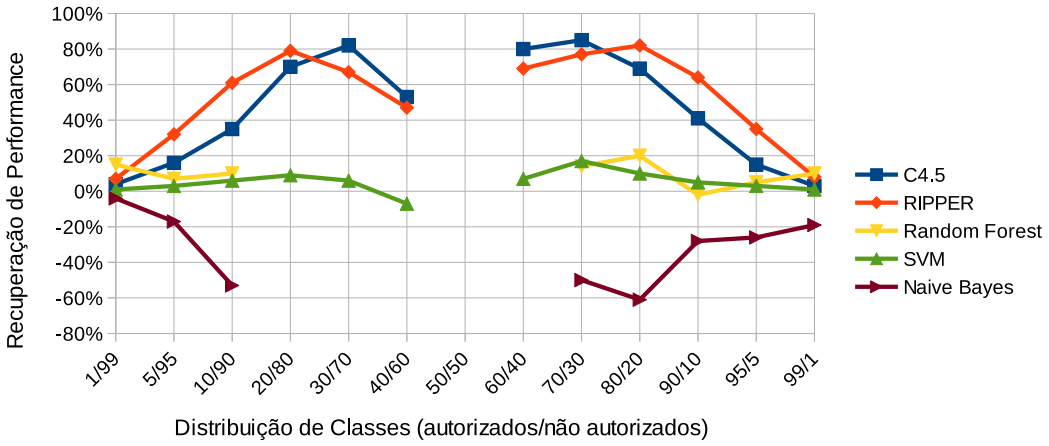

Figura 7 – Valores médios de recuperação de performance obtidos pelo método MetaCost segundo a Equação 4.3.

A Tabela 7 e a Figura 7 mostram que a recuperação de performance obtida pela aplicação do método MetaCost é, de certa forma, semelhantes à obtida pelo método *Random Oversampling* considerando os algoritmos C4.5 e RIPPER. Para os algoritmos *Random Forest* e SVM os valores de recuperanção ficaram abaixo de 20%, indicando que estes algoritmos pouco se beneficiaram deste método de tratamento. Algo que chama a atenção na Figura 7 é o comportamento do algoritmo *Naive Bayes*, que obteve valores de recuperação de performance bem abaixo de zero, indicando que a performance após o tratamento com este método foi pior do que a obtida com os dados originais. O resultado negativo da combinação do método MetaCost com o algoritmo *Naive Bayes* pode estar relacionado ao desafio de reatribuir os rótulos descrito por Aggarwal (2014b): as probabilidades calculadas pelos combinação de classificadores e que serviram para a reatribuição dos rótulos, como determina este método de tratamento, podem não ter refletido as propriedades intrínsecas dos dados.

5.3 Intervalos de Confiança

Segundo o experimento descrito neste trabalho, as etapas de divisão das bases de dados, de medição da perda de performance e de aplicação dos métodos de tratamento

foram repetidas por 100 vezes. Essa repetição permite que os valores das equações 4.1 e 4.2, que representam a perda de performance registrada nos dados originais e tratados, respectivamente, sejam analisados por meio de suas médias e respectivos intervalos de confiança, como mostrados nas figuras 8 e 9 e suas respectivas tabelas 8 e 9.

Nas figuras 8 e 9, o valor médio da perda de performance é representado por um ponto e o intervalo de confiança por um segmento de reta, indicando o menor e maior valor que pode ser alcançado pela média. Se um dado intervalo não contém o valor 0, significa que há diferença significativa entre a performance medida pela métrica AUC em uma dada distribuição de classe comparada com a performance medida na distribuição 50/50 (distribuição de classes tomada como referência). Também, se dois intervalos de confiança se sobrepõem então não há diferença significativa entre eles.

A Figura 8 apresenta os valores médios de perda de performance agrupados por distribuição de classe, registrados com base nos dados originais e nos dados tratados por cada um dos métodos estudados. Por essa figura é possível observar o formato da letra "U" rotacionada 90º no sentido horário, indicando que os valores mais baixos de perda de performance são aqueles em que a distribuição de classes é mais próxima de 50/50. Essa observação vale tanto para os dados originais quanto para os dados tratados e reforça a concepção de que, de um modo geral, há uma relação entre o desbalanceamento de classes e a perda de performance dos algoritimos.

Quanto aos métodos de tratamento, a Figura 8 mostra que o método *Random Oversampling* apresentou vantagem significativa quando comparado aos demais, principalmente nas distribuições de classe entre 70/30 e 99/1, em que a classe majoritária representam os serviços autorizados. Esta observação é importante devido ao fato de que, em comparação a outros métodos como o SMOTE, o método *Random Oversampling* possui baixo consumo de recursos computacionais.

Sobre o método SMOTE, nota-se pela Figura 8 que a perda de performance aumentou após o tratamento com este método naquelas bases de dados em que a classe minoritária representa os serviços não autorizados. Esse resultado pode estar relacionado à presença de exemplares ruidosos ou sobreposição de classes (*overlapping*), duas características dos dados que podem ter sido potencializadas com a aplicação deste método (SáEZ et al., 2015).

A Figura 9 também apresenta os valores médios de perda de performance registrados com base nos dados originais e tratados, agora agrupando-se por algoritmo de classificação. Essa figura mostra que os algoritimos C4.5, SVM e RIPPER foram aqueles que apresentaram a maior perda de performance quando aplicados aos dados originais (não tratados por nenhum método). Estes mesmos algoritmos obtiveram perdas de performance bem menores após os dados terem sido tratados. Já o algorítimo *Naive Bayes* apresentou um aumento da perda de performance para cada método de tratamento, significando que

a performance medida com base na métrica AUC apresentou valores menores nas bases de dados tratadas do que nas bases de dados originais (não tratadas). Novamente, a Figura 9 mostra a superioridade do método *Random Oversampling* sobre os demais, principalmente para os algoritmos C4.5, RIPPER e SVM.

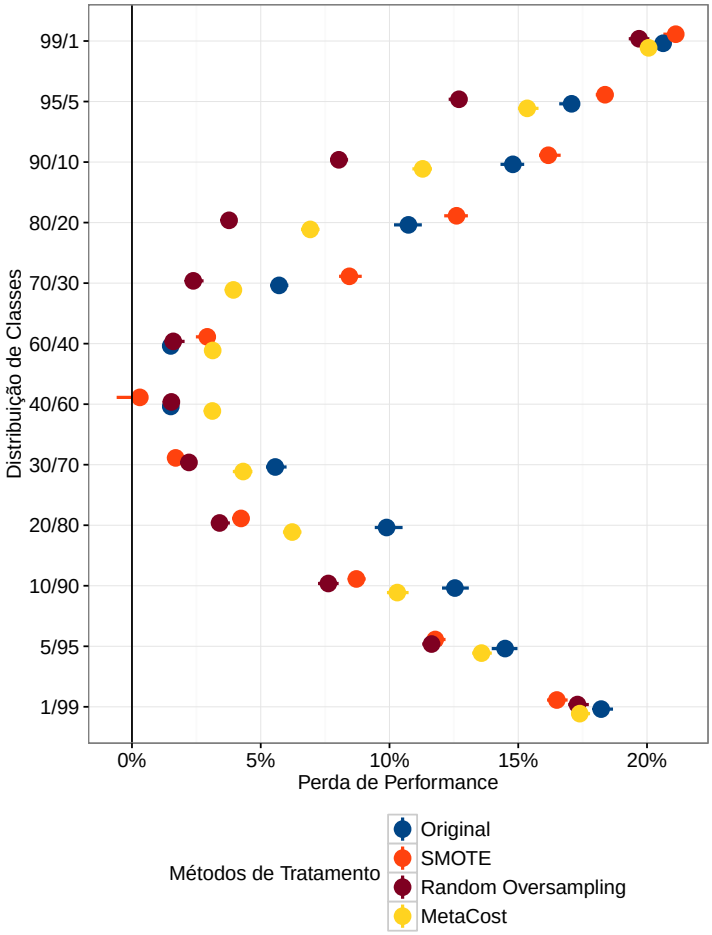

Figura 8 – Intervalos de confiança para a perda de performance segundo os níveis de balanceamento.

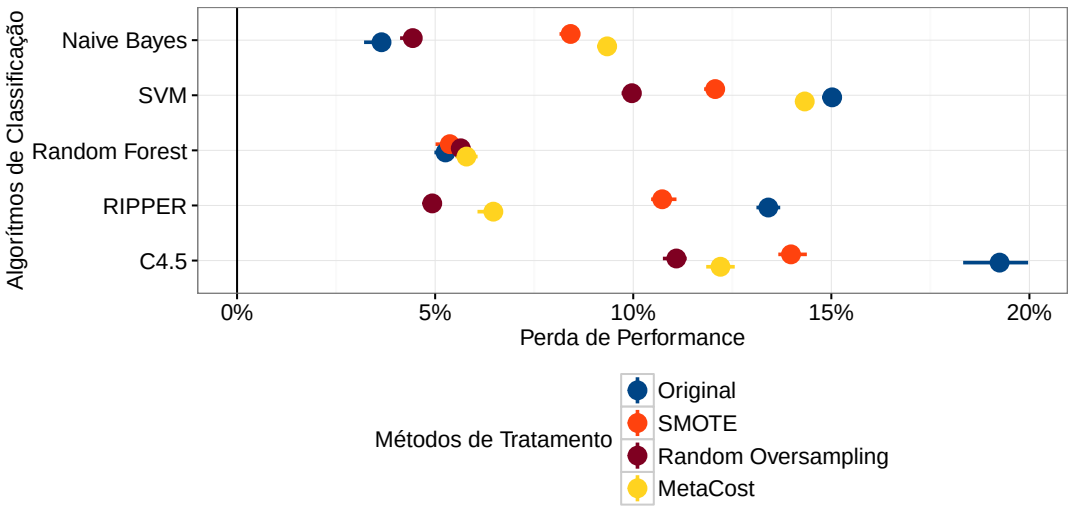

Figura 9 – Intervalos de confiança para a perda de performance segundo os algoritmos de classificação.

Tabela 8 – Médias e intervalos de confiança para a perda de performance, por nível de balanceamento e método de tratamento.

Distribuição de Classes	Método de Tratamento	Média	Mínimo	Máximo
1/99	Original	0.1821967312	0.1799967092	0.1867967772
1/99	SMOTE	0.164974181	0.161174143	0.169174223
1/99	Random Oversampling	0.1730050026	0.1694049666	0.1774050466
1/99	MetaCost	0.1738809403	0.1710809123	0.1778809803
5/95	Original	0.1448981801	0.1396981281	0.1496982281
5/95	SMOTE	0.1177619518	0.1141619158	0.1217619918
5/95	Random Oversampling	0.1163328477	0.1127328117	0.1203328877
5/95	MetaCost	0.1357276555	0.1321276195	0.1397276955
10/90	Original	0.1253944208	0.1203943708	0.1307944748
10/90	SMOTE	0.0871361162	0.0839360842	0.0907361522
10/90	Random Oversampling	0.0762485637	0.0722485237	0.0802486037
10/90	MetaCost	0.1030153553	0.0990153153	0.1074153993
20/80	Original	0.0988794025	0.0942793565	0.1050794645
20/80	SMOTE	0.0423549783	0.0391549463	0.0455550103
20/80	Random Oversampling	0.0340140517	0.0312140237	0.0380140917
20/80	MetaCost	0.0622181594	0.0586181234	0.0658181954
30/70	Original	0.0556353429	0.0520353069	0.0600353869
30/70	SMOTE	0.0169578075	0.0157577955	0.0221578595
30/70	Random Oversampling	0.0221259892	0.0169259372	0.0233260012
30/70	MetaCost	0.0431668269	0.0391667869	0.0467668629
40/60	Original	0.015069542	0.01386953	0.01586955
40/60	SMOTE	0.0030468696	-0.0059532204	0.0060468996
40/60	Random Oversampling	0.0152913994	0.0144913914	0.0156914034
40/60	MetaCost	0.0312143495	0.0292143295	0.0340143775
60/40	Original	0.015052444	0.014252436	0.015452448
60/40	SMOTE	0.0292248443	0.0248248003	0.0308248603
60/40	Random Oversampling	0.01603125	0.015231242	0.020431294
60/40	MetaCost	0.0313402248	0.0297402088	0.0341402528
70/30	Original	0.0571223014	0.0539222694	0.0607223374
70/30	SMOTE	0.0844143209	0.0804142809	0.0892143689
70/30	Random Oversampling	0.0238055409	0.0226055289	0.0278055809
70/30	MetaCost	0.0393794325	0.0369794085	0.0425794645
80/20	Original	0.1073555904	0.1017555344	0.1125556424
80/20	SMOTE	0.1260401149	0.1212400669	0.1304401589
80/20	Random Oversampling	0.0377061155	0.0353060915	0.0397061355
80/20	MetaCost	0.0692233353	0.0656232993	0.0728233713
90/10	Original	0.1479039675	0.1431039195	0.1523040115
90/10	SMOTE	0.1616691292	0.1580690932	0.1664691772
90/10	Random Oversampling	0.0803219047	0.0771218727	0.0839219407
90/10	MetaCost	0.1128886028	0.1088885628	0.1164886388
95/5	Original	0.1707048925	0.1659048445	0.1739049245
95/5	SMOTE	0.1837006135	0.1801005775	0.1869006455
95/5	Random Oversampling	0.1269778388	0.1229777988	0.1301778708
95/5	MetaCost	0.1534668958	0.1498668598	0.1578669398
99/1	Original	0.2062619315	0.2018618875	0.2090619595
99/1	SMOTE	0.2111553329	0.2063552849	0.2127553489
99/1	Random Oversampling	0.1969103058	0.1929102658	0.2009103458
99/1	MetaCost	0.20061043	0.198410408	0.206010484

Tabela 9 – Médias e intervalos de confiança para a perda de performance, por algoritmos e método de tratamento.

Algoritmo	Método de Tratamento	Média	Mínimo	Máximo
C4.5	Original	0.1924743677	0.1832742757	0.1996744397
C4.5	SMOTE	0.1398189851	0.1366189531	0.1438190251
C4.5	Random Oversampling	0.1108834954	0.1074834614	0.1134835214
C4.5	MetaCost	0.1220191465	0.1184191105	0.1256191825
RIPPER	Original	0.134094641	0.131094611	0.137094671
RIPPER	SMOTE	0.1073115297	0.1045115017	0.1109115657
RIPPER	Random Oversampling	0.0492655077	0.0472654877	0.0516655317
RIPPER	MetaCost	0.0646891308	0.0606890908	0.0662891468
Random Forest	Original	0.0526091266	0.0498090986	0.0558091586
Random Forest	SMOTE	0.0537043056	0.0501042696	0.0569043376
Random Forest	Random Oversampling	0.0564680781	0.0536680501	0.0588681021
Random Forest	MetaCost	0.0579124381	0.0551124101	0.0607124661
SVM	Original	0.1502068808	0.1478068568	0.1518068968
SVM	SMOTE	0.1206872913	0.1178872633	0.1230873153
SVM	Random Oversampling	0.0996548163	0.0970547903	0.1010548303
SVM	MetaCost	0.1432855164	0.1408854924	0.1448855324
Naive Bayes	Original	0.0364575735	0.0320575295	0.0380575895
Naive Bayes	SMOTE	0.0841902281	0.0813902001	0.0865902521
Naive Bayes	Random Oversampling	0.0443476176	0.0411475856	0.0463476376
Naive Bayes	MetaCost	0.0934157788	0.0910157548	0.0958158028

6 Conclusão e Trabalhos Futuros

Este trabalho procurou investigar os efeitos do desbalanceamento de classes na aprendizagem da regulação de planos de saúde. Para isso, seguiu-se o experimento proposto por Prati, Gustavo E A P e Silva (2014) utilizando 3 bases de dados originadas dos serviços de regulação de duas operadoras de planos de saúde distintas. Uma dessas operadoras atua na cidade de Teresina, Piauí e do seu serviço de regulação foram construídas 2 bases de dados: uma contendo apenas solicitação de serviços odontológicos e a outra contendo apenas solicitação de serviços ambulatoriais. A outra operadora atua na cidade de Recife, Pernambuco e do seu serviço de regulação foi construída uma única base de dados contendo solicitações de serviços odontológicos.

6.1 Conclusão

A execução do experimento que investigou os efeitos do desbalanceamento ocorreu em etapas. Na primeira delas, cada uma das três bases de dados foi dividida em bases de dados menores destinadas ao treinamento e teste de classificadores. As bases destinadas ao treinamento dos classificadores possuem a distribuição de classes modificada artificialmente para abranger o espectro que vai desde 1/99 até 99/1. Já as bases destinadas aos testes possuem a mesma distribuição de classes da base da qual foi originada.

De posse das bases de dados divididas, seguiu-se para a etapa seguinte em que classificadores de diferentes algoritmos foram treinados e testados. Com os resultados dos testes calculou-se os valores de perda de performance, que comparam a performance obtida em uma dada distribuição de classes com as performances obtidas na distribuição 50/50, que o experimento assume como valor de referência.

Com os valores de perda de performance calculados ficou claro o que já era esperado: que o desbalanceamento de classes afeta sim a aprendizagem da regulação. Mais precisamente, notou-se que os algoritmos C4.5, RIPPER e SVM foram os mais afetados ao passo que os algoritmos *Random Forest* e *Naive Bayes* foram os menos afetados. Estes dois ultimos inclusive apresentaram valores constantes de AUC no intervalo de distribuição de classes entre 20/80 e 50/50.

A etapa seguinte foi aquela em que os métodos de tratamento para desbalanceamento de classe foram aplicados às bases de dados de treinamento. Dessa forma foi possível construir novos classificadores utilizando-se dos dados tratados e em seguida a performance foi medida novamente nas mesmas bases de dados de teste. Com os resultados dessa etapa foi possível calcular a perda de performance dos classificadores treinados com os dados

tratados e também calcular o quanto da performance de predição foi recuperada por conta da utilização dos métodos de tratamento.

Com as medidas de recuperação de performance notou-se que os algoritmos C4.5, RIPPER e SVM (os que apresentaram as maiores perdas de performance) foram também os que mais se beneficiaram do método *Random Oversampling*. Já os classificadores construídos com o *Random Forest* e *Naive Bayes* alcançaram uma recuperação pequena ou mesmo aumentaram a perda de performance em algumas distribuições de classes, considerando esse mesmo método de tratamento.

Com relação ao método SMOTE, o que ficou observado é que a recuperação de performance obtida nas distribuições em que a classe minoritária representa os serviços que foram autorizados na regulação foi bem maior do que a recuperação obtida nas distribuições inversas (aquelas em que a classe minoritária representa os serviços não autorizados). Esse comportamento assimétrico pode estar relacionado com as características negativas desse método comentadas por Sáez et al. (2015): por considerar apenas medidas de distância, o método SMOTE pode multiplicar exemplares ruidosos ou mesmo aumentar a sobreposição entre as classes. Logo, o comportamento assimétrico observado no experimento pode estar relacionado a presença de dados ruidosos ou sobreposição de classes nas bases de dados empregadas neste estudo.

Sobre o método MetaCost, este proporcionou uma recuperação de performance aos algoritmos C4.5 e RIPPER semelhante àquela proporcionada pelo método *Random Oversampling*. Para os algoritmos *Random Forest* e SVM a recuperação foi bem menor. O resultado que deve-se destacar é que a performance dos classificadores treinados com o *Naive Bayes* foi das piores comparando-se com os demais algoritmos. Tanto o *Naive Bayes* quanto o MetaCost são baseados em probabilidades e uma provável explicação para a o ruím desempenho dessa combinação é que a reatribuição de rótulos executada por esse método de tratamento pode ter divergido bastante das propriedades intrínsecas dos dados Aggarwal (2014b).

Além das interpretações comentadas nos parágrafos anteriores, as diversas repetições das etapas que mediram a perda de performance antes e depois do tratamento dos dados (100 vezes exatamente) permitiram que os valores de perda de performance pudessem ser analisados por meio de intervalos de confiança e agrupados por distribuição de classes e algoritmo de classificação. Dessa forma, foi possível observar que o método de tratamento *Random Oversampling* foi semelhante ou superior aos demais para a grande maioria das distribuições de classes estudadas. Essa observação é importante pois esse método consome bem menos recursos computacionais comparando-se ao SMOTE e ao MetaCost.

6.2 Continuidade da Pesquisa

Com o objetivo de apoiar o desenvolvimento de sistemas que melhorem a execução do processo de regulação ou mesmo outros processos ligados a gestão de planos de saúde, este trabalho pode ser continuado ou expandido das seguintes formas:

- estudo de métodos de tratamento para desbalanceamento de classes mais sofisticados,
- estudo de *ensembles* para o aprendizado da regulação,
- estudo sobre outras características dos dados que também afetam a aprendizagem.

Pelos resultados encontrados notou-se que o método *Random Oversampling* (que é o método mais simples dos três estudados) obteve vantagens semelhantes ou superiores aos métodos SMOTE e MetaCost, que são mais complexos e custosos computacionamente. Logo, há importância em pesquisar por outros métodos de tratamento, como por exemplo variações do método SMOTE, que possam proporcionar maiores recuperações de performance.

Também pelos resultados encontrados, notou-se a superioridade do algoritmo *Random Forest* no aprendizado da regulação. Este algoritmo, diferente dos demais estudados, constrói um *ensemble* ou combinação de classificadores baseados em árvores de decisão. Os classificadores treinados com esse algoritmo atingiram maiores valores de AUC e também foram mais tolerantes quanto ao desbalanceamento de classes. Dessa forma, também é importante pesquisar por outros tipos de *ensemble* que possam ser tolerantes ao desbalanceamento de classes e talvez superiores ao Random Forest quanto a performance na aprendizagem da regulação.

Por fim, já afastando-se daquilo que foi desenvolvido neste trabalho, pode-se pesquisar por outros problemas relacionados a características intrínsecas dos dados que também afetam a aprendizagem. Algumas desses problemas estão listados abaixo (LóPEZ et al., 2013):

- Pequenas Disjunções (do inglês *Small Disjuncts*): situação que ocorre quando os conceitos (classes) são representados dentro de grupos pequenos nos dados. Embora essa característica dos dados seja comum, em combinação com desbalanceamento de classes torna-se um problema maior;
- Falta de Densidade (do inglês *Lack of Density*): situação que ocorre quando a quantidade de dados é insuficiente para que os algoritmos de classificação obtenham uma boa generalização;

- Sobreposição (do inglês *Overlapping*): ocorre quando, em uma determinada região do espaço de dados, há quantidades semelhantes de exemplares de cada classe, tornando difícil ou mesmo impossível para um algoritmo de classificação distinguir corretamente;

- Ruidos: em combinação com o desbalanceamento de classes, a presença de dados ruidosos pertencentes a classe minoritária torna ainda mais difícil a tarefa de aprendizagem;

- Diferença na distribuição de classes entre bases: ocorre quando há diferença na distribuição de classes entre os dados utilizados no treinamento e teste dos algoritmos. Em domínios em que o desbalanceamento de classes é acentuado, essa diferença pode ter um impacto muito grande na performance medida nos testes ou mesmo quando o sistema for colocado em produção.

Referências

AGGARWAL, C. C. An introduction to data classification. In: AGGARWAL, C. C. (Ed.). *Data Classification: Algorithms and Applications*. CRC Press, 2014. p. 1–36. Disponível em: <http://www.crcnetbase.com/doi/abs/10.1201/b17320-2>. Citado na página 19.

AGGARWAL, C. C. Rare class learning. In: *Data Classification: Algorithms and Applications*. [s.n.], 2014. p. 445–468. Disponível em: <http://www.crcnetbase.com/doi/abs/10.1201/b17320-18>. Citado 3 vezes nas páginas 29, 43 e 50.

ALBISUA, I. n. et al. The quest for the optimal class distribution: an approach for enhancing the effectiveness of learning via resampling methods for imbalanced data sets. *Prog Artif Intell*, Springer-Verlag, v. 2, n. 1, p. 45–63, 24 nov. 2012. Citado 4 vezes nas páginas 12, 23, 33 e 34.

AMAZON EC2. 2016. Acessado em 13 de agosto de 2016. Disponível em: <https://aws.amazon.com/pt/ec2/>. Citado na página 37.

ANS. *Dados e Indicadores do Setor de Saúde Suplementar*. 2015. Acesso em 8 de março de 2016. Disponível em: <http://www.ans.gov.br/perfil-do-setor/dados-e-indicadores-do-setor>. Citado na página 3.

ARAÚJO, F. H. D. de; SANTANA, A. M.; NETO, P. d. A. dos S. Uma abordagem influenciada por pré-processamento para aprendizagem do processo de regulação médica. *J. Health Inform. Dev. Ctries*, v. 7, n. 1, 14 mar. 2015. Citado 3 vezes nas páginas 8, 61 e 65.

BATISTA, G. E. A. P. A.; PRATI, R. C.; MONARD, M. C. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 6, n. 1, p. 20–29, jun. 2004. Citado na página 34.

BOLóN-CANEDO, V.; nO, N. S.-M.; ALONSO-BETANZOS, A. A review of feature selection methods on synthetic data. *Knowl. Inf. Syst.*, Springer-Verlag, v. 34, n. 3, p. 483–519, 30 mar. 2012. Citado na página 25.

BOLóN-CANEDO, V. et al. A review of microarray datasets and applied feature selection methods. *Inf. Sci.*, v. 282, p. 111–135, 20 out. 2014. Citado na página 26.

BREIMAN, L. Random forests. *Mach. Learn.*, Kluwer Academic Publishers, v. 45, n. 1, p. 5–32, 2001. Citado na página 24.

BROWN, I.; MUES, C. An experimental comparison of classification algorithms for imbalanced credit scoring data sets. *Expert Syst. Appl.*, v. 39, n. 3, p. 3446–3453, 15 fev. 2012. Citado na página 24.

CHANDOLA, V.; SUKUMAR, S. R.; SCHRYVER, J. C. Knowledge discovery from massive healthcare claims data. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2013. (KDD '13), p. 1312–1320. Citado na página 4.

CHAWLA, N. V. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, v. 16, p. 321–357, 2002. Citado na página 28.

CHAWLA, N. V. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.*, v. 16, p. 321–357, 2002. Citado na página 34.

COHEN, W. W. Fast effective rule induction. In: *In Proceedings of the Twelfth International Conference on Machine Learning*. [S.l.: s.n.], 1995. Citado na página 23.

DAS, B.; KRISHNAN, N. C.; COOK, D. J. Handling imbalanced and overlapping classes in smart environments prompting dataset. In: *Data Mining for Service*. [S.l.]: Springer Berlin Heidelberg, 2014, (Studies in Big Data). p. 199–219. Citado na página 14.

DENG, H. et al. Probabilistic models for classification. In: AGGARWAL, C. C. (Ed.). *Data Classification: Algorithms and Applications*. CRC Press, 2014. p. 65–86. Disponível em: <http://www.crcnetbase.com/doi/abs/10.1201/b17320-4>. Citado na página 24.

DESSÌ, N.; PES, B. Similarity of feature selection methods: An empirical study across data intensive classification tasks. *Expert Syst. Appl.*, v. 42, n. 10, p. 4632–4642, 15 jun. 2015. Citado na página 26.

DITTMAN, D. J.; KHOSHGOFTAAR, T. M.; NAPOLITANO, A. The effect of data sampling when using random forest on imbalanced bioinformatics data. In: *Information Reuse and Integration (IRI), 2015 IEEE International Conference on*. [S.l.: s.n.], 2015. p. 457–463. Citado 2 vezes nas páginas 14 e 24.

DOMINGOS, P. MetaCost: A general method for making classifiers cost-sensitive. In: *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 1999. (KDD '99), p. 155–164. Citado 2 vezes nas páginas 29 e 34.

DUA, P.; BAIS, S. Supervised learning methods for fraud detection in healthcare insurance. In: *Machine Learning in Healthcare Informatics*. [S.l.]: Springer Berlin Heidelberg, 2014, (Intelligent Systems Reference Library). p. 261–285. Citado 2 vezes nas páginas 4 e 5.

FARQUAD, M. A. H.; BOSE, I. Preprocessing unbalanced data using support vector machine. *Decis. Support Syst.*, v. 53, n. 1, p. 226–233, abr. 2012. Citado 2 vezes nas páginas 23 e 25.

FAWCETT, T. An introduction to ROC analysis. *Pattern Recognit. Lett.*, Elsevier Science Inc., New York, NY, USA, v. 27, n. 8, p. 861–874, jun. 2006. Citado na página 21.

FERNáNDEZ, A.; GARCÍA, S.; HERRERA, F. Addressing the classification with imbalanced data: Open problems and new challenges on class distribution. In: *Hybrid Artificial Intelligent Systems*. [S.l.]: Springer Berlin Heidelberg, 2011, (Lecture Notes in Computer Science). p. 1–10. Citado na página 27.

GALAR, M. et al. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and Hybrid-Based approaches. *IEEE Trans. Syst. Man Cybern. C Appl. Rev.*, v. 42, n. 4, p. 463–484, jul. 2012. Citado na página 28.

GARCÍA, V.; SáNCHEZ, J. S.; MOLLINEDA, R. A. On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. *Knowledge-Based Systems*, v. 25, n. 1, p. 13–21, fev. 2012. Citado 2 vezes nas páginas 11 e 12.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, JMLR.org, v. 3, p. 1157–1182, mar. 2003. Citado na página 25.

HALL, M. et al. The WEKA data mining software: An update. *SIGKDD Explor. Newsl.*, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, nov. 2009. Citado 2 vezes nas páginas 33 e 65.

HAN, J.; KAMBER, M.; PEI, J. *Data Mining: Concepts and Techniques*. 3rd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN 0123814790, 9780123814791. Citado 3 vezes nas páginas 17, 19 e 21.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. Unsupervised learning. In: _____. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer New York, 2009. p. 485–585. ISBN 978-0-387-84858-7. Disponível em: <http://dx.doi.org/10.1007/978-0-387-84858-7_14>. Citado na página 18.

HE, H.; GARCIA, E. A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.*, ieeexplore.ieee.org, v. 21, n. 9, p. 1263–1284, 2009. Citado 3 vezes nas páginas 27, 28 e 32.

HE, Z. et al. Predicting Drug-Target interaction networks based on functional groups and biological features. *PLoS One*, Public Library of Science, v. 5, n. 3, p. e9603, 3 nov. 2010. Citado na página 65.

HILLERMAN, T. P.; CARVALHO, R. N.; REIS, A. C. B. Analyzing suspicious medical visit claims from individual healthcare service providers using K-Means clustering. In: *Electronic Government and the Information Systems Perspective*. [S.l.]: Springer International Publishing, 2015, (Lecture Notes in Computer Science). p. 191–205. Citado 3 vezes nas páginas 3, 5 e 7.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: Trends, perspectives, and prospects. *Science*, science.sciencemag.org, v. 349, n. 6245, p. 255–260, 17 jul. 2015. Citado na página 18.

KELLEY, R. R. *Where can $700 Billion in Waste be cut annually from the US Healthcare System?* [S.l.]: Thomson Reuters, 2009. Citado na página 4.

KOSE, I.; GOKTURK, M.; KILIC, K. An interactive machine-learning-based electronic fraud and abuse detection system in healthcare insurance. *Appl. Soft Comput.*, Elsevier, v. 36, p. 283–299, nov. 2015. Citado 2 vezes nas páginas 4 e 5.

LEE, V. E.; LIU, L.; JIN, R. Decision trees: Theory and algorithms. In: *Data Classification: Algorithms and Applications*. [s.n.], 2014. p. 87–120. Disponível em: <http://www.crcnetbase.com/doi/abs/10.1201/b17320-5>. Citado na página 23.

LI, B.-Q. et al. Prediction of protein domain with mRMR feature selection and analysis. *PLoS One*, v. 7, n. 6, p. e39308, 15 jun. 2012. Citado na página 26.

LI, Y. et al. Ensemble learning. In: AGGARWAL, C. C. (Ed.). *Data Classification: Algorithms and Applications*. CRC Press, 2014. p. 483–510. Disponível em: <http://www.crcnetbase.com/doi/abs/10.1201/b17320-20>. Citado na página 25.

LIBBRECHT, M. W.; NOBLE, W. S. Machine learning applications in genetics and genomics. *Nat. Rev. Genet.*, nature.com, v. 16, n. 6, p. 321–332, jun. 2015. Citado na página 18.

LIU, X.-Y.; WU, J.; ZHOU, Z.-H. Exploratory undersampling for class-imbalance learning. *IEEE Trans. Syst. Man Cybern. B Cybern.*, v. 39, n. 2, p. 539–550, abr. 2009. Citado na página 33.

LóPEZ, V. et al. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Inf. Sci.*, v. 250, p. 113–141, 20 nov. 2013. Citado 2 vezes nas páginas 27 e 51.

LóPEZ, V.; FERNáNDEZ, A.; HERRERA, F. On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed. *Inf. Sci.*, v. 257, p. 1–13, 1 fev. 2014. Citado 2 vezes nas páginas 21 e 22.

LORENA, A. C. et al. Comparing machine learning classifiers in potential distribution modelling. *Expert Syst. Appl.*, v. 38, n. 5, p. 5268–5275, maio 2011. Citado na página 23.

MINIMUM Redundancy Maximum Relevance Feature Selection. 2016. Acesso em 21 de junho de 2016. Disponível em: <http://penglab.janelia.org/proj/mRMR>. Citado na página 65.

MORRIS, L. Combating fraud in health care: an essential component of any cost containment strategy. *Health Affairs*, v. 28, n. 5, p. 1351–1356, set. 2009. Citado na página 4.

MRMR. *minimum Frequently Asked Questions*. 2016. Acesso em 21 de junho de 2016. Disponível em: <http://penglab.janelia.org/proj/mRMR/FAQ_mrmr.htm>. Citado na página 26.

PENG, H.; LONG, F.; DING, C. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Pattern Anal. Mach. Intell.*, v. 27, n. 8, p. 1226–1238, ago. 2005. Citado na página 26.

PERNER, P.; APTE, C. Empirical evaluation of feature subset selection based on a real-world data set. *Eng. Appl. Artif. Intell.*, v. 17, n. 3, p. 285–288, 2004. Citado na página 23.

PRATI, R. C.; BATISTA, G. E. A. P. A.; MONARD, M. C. A survey on graphical methods for classification predictive performance evaluation. *IEEE Trans. Knowl. Data Eng.*, v. 23, n. 11, p. 1601–1618, nov. 2011. Citado na página 19.

PRATI, R. C.; Gustavo E A P; SILVA, D. F. Class imbalance revisited: a new experimental setup to assess the performance of treatment methods. *Knowl. Inf. Syst.*, Springer London, v. 45, n. 1, p. 247–270, 17 out. 2014. Citado 7 vezes nas páginas 6, 11, 31, 35, 37, 40 e 49.

QUINLAN, J. R. *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1993. ISBN 1-55860-238-0. Citado 2 vezes nas páginas 23 e 26.

RATHORE, S.; IFTIKHAR, M. A.; HUSSAIN, M. A novel approach for automatic gene selection and classification of gene based colon cancer datasets. In: *Emerging Technologies (ICET), 2014 International Conference on*. [S.l.: s.n.], 2014. p. 42–47. Citado na página 65.

ROKACH, L.; MAIMON, O. Classification trees. In: _____. *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer US, 2010. p. 149–174. ISBN 978-0-387-09823-4. Disponível em: <http://dx.doi.org/10.1007/978-0-387-09823-4_9>. Citado na página 26.

ROKACH, L.; MAIMON, O. Supervised learning. In: _____. *Data Mining and Knowledge Discovery Handbook*. Boston, MA: Springer US, 2010. p. 133–147. ISBN 978-0-387-09823-4. Disponível em: <http://dx.doi.org/10.1007/978-0-387-09823-4_8>. Citado 2 vezes nas páginas 18 e 19.

SáEZ, J. A. et al. SMOTE–IPF: Addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering. *Inf. Sci.*, Elsevier, v. 291, p. 184–203, 2015. Citado 4 vezes nas páginas 28, 42, 44 e 50.

SEIFFERT, C. et al. An empirical study of the classification performance of learners on imbalanced and noisy software quality data. *Inf. Sci.*, v. 259, p. 571–595, 20 fev. 2014. Citado 2 vezes nas páginas 13 e 21.

SIERS, M. J.; ISLAM, M. Z. Software defect prediction using a cost sensitive decision forest and voting, and a potential solution to the class imbalance problem. *Inf. Syst.*, Elsevier, v. 51, p. 62–71, 2015. Citado na página 14.

SOBHANI, P.; VIKTOR, H.; MATWIN, S. Learning from imbalanced data using ensemble methods and Cluster-Based undersampling. In: *New Frontiers in Mining Complex Patterns*. [S.l.]: Springer International Publishing, 2014, (Lecture Notes in Computer Science). p. 69–83. Citado na página 28.

STEFANOWSKI, J. Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data. In: *Emerging Paradigms in Machine Learning*. [S.l.]: Springer Berlin Heidelberg, 2013, (Smart Innovation, Systems and Technologies). p. 277–306. Citado 3 vezes nas páginas 6, 8 e 32.

USE WEKA in your Java code. 2016. Acesso em 21 de junho de 2016. Disponível em: <https://weka.wikispaces.com/Use+WEKA+in+your+Java+code#Filter-Filteringon-the-fly>. Citado 2 vezes nas páginas 33 e 34.

VERBIEST, N.; VERMEULEN, K.; TEREDESAI, A. Evaluation of classification methods. In: AGGARWAL, C. C. (Ed.). *Data classification : algorithms and applications*. [S.l.]: CRC Press, 2014, (Chapman & Hall/CRC Data Mining and Knowledge Discovery Series). ISBN 9781466586741. Citado 2 vezes nas páginas 21 e 22.

VERGARA, J. R.; ESTéVEZ, P. A. A review of feature selection methods based on mutual information. *Neural Comput. Appl.*, Springer London, v. 24, n. 1, p. 175–186, 13 mar. 2013. Citado na página 25.

WANG, P.; LIN, C. Support vector machines. In: AGGARWAL, C. C. (Ed.). *Data Classification: Algorithms and Applications*. CRC Press, 2014. p. 187–204. Disponível em: <http://www.crcnetbase.com/doi/abs/10.1201/b17320-8>. Citado na página 24.

WEISS, G. M.; PROVOST, F. Learning when training data are costly: the effect of class distribution on tree induction. *J. Artif. Intell. Res.*, jair.org, p. 315–354, 2003. Citado 2 vezes nas páginas 12 e 38.

WITTEN, I. H.; FRANK, E.; HALL, M. A. *Data Mining: Practical Machine Learning Tools and Techniques*. 3rd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011. ISBN 0123748569, 9780123748560. Citado 3 vezes nas páginas 17, 18 e 29.

WU, J. et al. Self-adaptive attribute weighting for naive bayes classification. *Expert Syst. Appl.*, v. 42, n. 3, p. 1487–1502, 15 fev. 2015. Citado na página 24.

WU, X. et al. Top 10 algorithms in data mining. *Knowl. Inf. Syst.*, Springer-Verlag, v. 14, n. 1, p. 1–37, 4 dez. 2007. Citado 2 vezes nas páginas 23 e 24.

XIAO, J. et al. Dynamic classifier ensemble model for customer classification with imbalanced class distribution. *Expert Syst. Appl.*, v. 39, n. 3, p. 3668–3675, 15 fev. 2012. Citado na página 14.

ZHANG, M. L.; ZHOU, Z. H. A review on Multi-Label learning algorithms. *IEEE Trans. Knowl. Data Eng.*, ieeexplore.ieee.org, v. 26, n. 8, p. 1819–1837, ago. 2014. Citado na página 18.

Apêndices

APÊNDICE A – Construção da Base de Dados Teresina Odontológico

Os que compõem a base Teresina Odontológico são originados do serviço de regulação odontológica de uma operadora de planos de saúde sem fins lucrativos que atua na cidade de Teresina, Piauí. Estes dados representam o atendimento oferecido por essa operadora a 4114 usuários (beneficiários), que no período compreendido entre os meses de abril de 2014 a março de 2015 foram atendidos por 70 diferentes dentistas. Como consequência do atendimento prestado, esses dentistas realizaram cerca de 28042 solicitações de serviços odontológicos, incluindo exames e outros tipos de procedimentos como restaurações e limpezas. Essas solicitações, ao serem submetidas ao serviço de regulação, foram analisadas por uma equipe de 8 profissionais e, por conseguinte, tiveram sua realização autorizada ou não segundo critérios técnicos pré-estabelecidos.

A.1 Definição dos Atributos

Inicialmente foram removidos dessa base de dados todos os atributos que assumiam valores constantes ou que não eram relevantes para o processo de regulação. Exemplos desses atributos são: periodicidade, duração de carências e idades mínimas e máximas para realização de consultas e exames. Também foram removidos aqueles que continham informações pessoais como nomes, documentos de identificação e endereços.

A definição da base de dados também empregou atributos previamente definidos em outro estudo sobre aprendizagem da regulação de planos de saúde e que também empregou dados sobre regulação odontológica (ARAÚJO; SANTANA; NETO, 2015). Esses atributos estão listados à seguir:

- quantidade de solicitações para um dado beneficiário no mesmo ano,
- quantidade de solicitações para um dado beneficiário no mesmo semestre,
- quantidade de solicitações para um dado beneficiário no mesmo mês,
- quantidade de solicitações de mesmo código para um dado beneficiário no mesmo ano,
- quantidade de solicitações de mesmo código para um dado beneficiário no mesmo semestre,

- quantidade de solicitações de mesmo código para um dado beneficiário no mesmo mês,
- quantidade de solicitações de mesma complexidade para um dado beneficiário por ano,
- quantidade de solicitações de mesma complexidade para um dado beneficiário por semestre,
- quantidade de solicitações de mesma complexidade para um dado beneficiário por mês,
- quantidade de guias solicitadas pelo mesmo profissional da solicitação por ano,
- quantidade de guias solicitadas pelo mesmo profissional da solicitação por semestre,
- quantidade de guias solicitadas pelo mesmo profissional da solicitação por mês,
- profissional solicitante,
- mês da solicitação,
- código do serviço,
- porte anestésico,
- valor,
- valor de moderação,
- idade do beneficiário,
- sexo do beneficiário,
- tipo do beneficiário,
- resultado da regulação.

A.1.1 Odontograma

Além dos já citados, procurou-se criar novos atributos utilizando-se informações que descrevem a situação bucal do beneficiário e que fazem parte dos dados enviados à operadora de planos de saúde no momento da solicitação dos serviços. Esse conjunto de informações é chamado de odontograma e funciona de forma análoga a um mapa, em que as regiões representam estruturas bucais (arcadas, quadrantes, dentes, faces, etc.) e servem basicamente a duas finalidades:

- representar a situação bucal por meio de um valor pré-definido associado às regiões que representam os dentes;

- indicar o local em que um determinado serviço será realizado, caso seja autorizado durante a regulação.

Os valores pré-definidos que representam a situação de um dado dente no odontograma são: “A” (ausente), “C” (cariado), “E” (indicado para extração), “H” (hígido), “IC” (indicado para canal), “IE” (indicado para endodontia), “R-CT” (restaurado coroa total), “R1/R2” (restaurado em uma ou duas faces), “R3/R4” (restaurado em 3 ou quatro faces), “NI” (não informado).

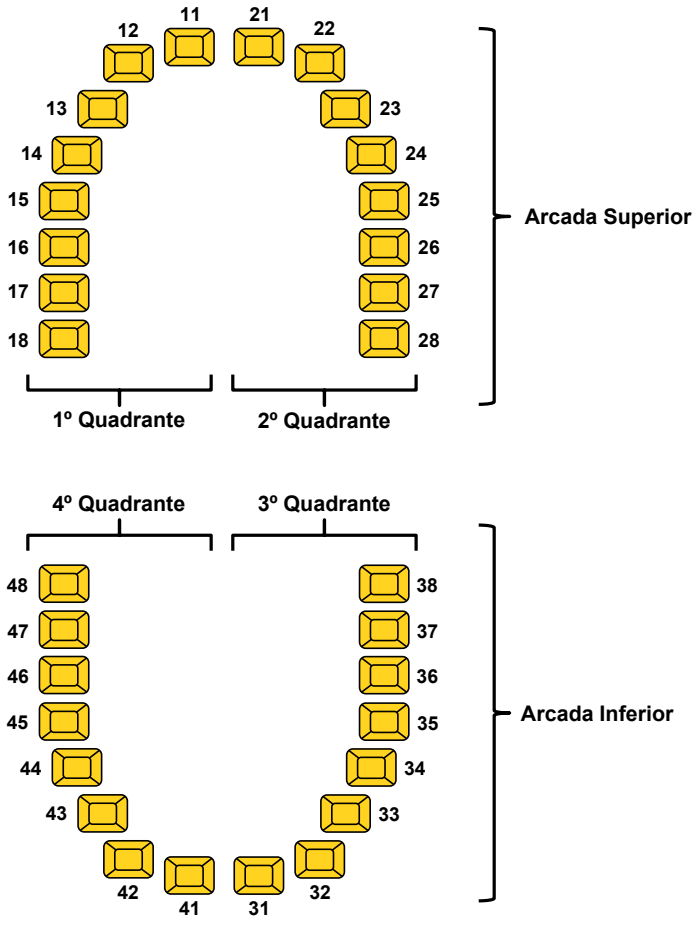

Figura 10 – Visão simplificada das estruturas bucais de um ser humano adulto que compõem um odontograma.

A Figura 10 mostra de forma simplificada parte das estruturas bucais de um ser humano adulto que compõem um odontograma. Nela é possível ver os dentes identificados por códigos numéricos e organizados em arcadas e quadrantes. O preenchimento do odontograma é feito da seguinte forma:

- anotando-se cada dente com um código que indique seu estado atual (“A” para dentes ausentes, “C” para dentes cariados, etc.);
- anotando-se em quais estruturas o serviço solicitado será aplicado.

Dessa forma, é possível informar que o serviço “Restauração” será aplicado à estrutura “Dente 17” que por sua vez está marcado com o código “C”, que indica que o referido dente está cariado. De posse dessas informações foram criados os seguintes conjuntos de atributos:

- Tipo 1 (10 atributos): quantidade de dentes em uma dada situação;
- Tipo 2 (52 atributos): situação atual dos dentes;
- Tipo 3 (73 atributos): estruturas em que um serviço deverá ser realizado (arcadas, quadrantes, dentes, etc.);
- Tipo 4 (1 atributo): tipo da restauração solicitada em dente anterior (para os casos que o serviço solicitado é uma restauração em um dente anterior).

Ao todo foram definidos 157 atributos com base no estudo anterior e nas informações contidas no odontograma (além do atributo que define se o serviço foi autorizado ou não durante a regulação), sendo que aqueles definidos à partir do odontograma representam a maioria desses atributos, como mostra a Figura 11.

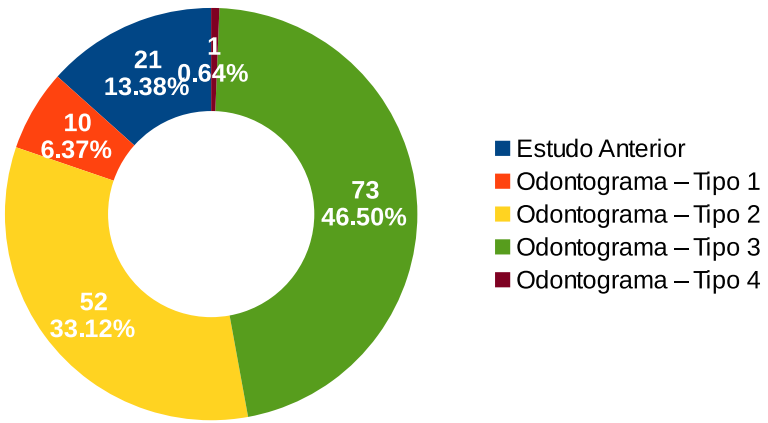

Figura 11 – Quantidades dos atributos definidos com base no estudo anterior e no odontograma.

A.2 Seleção de Atributos

De posse dos 158 atributos, decidiu-se selecionar um subconjunto desses atributos capaz de proporcionar um menor erro de generalização durante o treinamento dos algoritmos

de classificação e que também reduzisse os efeitos negativos causados por ruídos nos dados. Para isso, foi aplicado um procedimento conhecido como Seleção Incremental de Atributos (do inglês *Incremental Feature Selection*) (HE et al., 2010). Segundo esse procedimento, uma lista de N atributos é utilizada na construção de N subconjuntos de atributos, de acordo com a Expressão A.1:

$$S_i = \{f_1, f_2, ..., f_i\} \, (1 \leqslant i \leqslant N). \tag{A.1}$$

A Expressão A.1 pode ser exemplificada da seguinte forma: dada uma lista com 3 atributos, os subconjuntos possíveis são S_1 (formado pelo primeiro atributo), S_2 (formado pelo primeiro e segundo atributos) e S_3 (formado pelos três atributos).

Os sobconjuntos de atributos formados por meio da Expressão A.1 foram então avaliados utilizando-se a performance obtida por classificadores construídos com o algoritmo C4.5. Cada base de dados definida pelo subconjunto de atributos foi balanceada utilizando-se uma combinação dos métodos *Random Oversampling* e *Random Undersampling* para que atingissem a distribuição de classes 50/50. A avaliação da performance seguiu o procedimento de validação cruzada e a medida de performance selecionada foi a acurácia. Tanto o algoritmo C4.5 quanto a acurácia foram escolhidos devido a simplicidade, baixo custo computacional e facilidade de implementação por meio da API de programação disponibilizada pelo *WEKA* (HALL et al., 2009).

A seleção de atributos adotada na base de dados Teresina Odontológico empregou 3 listas com 157 atributos cada (excluiu-se o atributo alvo que representa o resultado da classificação). Um dos objetivos foi comparar os subconjuntos gerados a partir de cada lista e ao final escolher aquele de melhor qualidade segundo um critério pré-estabelecido. Outro objetivo foi também comparar os métodos utilizados na criação das listas de atributos.

A primeira das listas de atributos foi criada segundo o método da Razão de Ganho de Informação (RGI) utilizando-se a ferramenta *WEKA*. Esse método foi escolhido devido a facilidade de uso e por ser comumente citado na literatura, inclusive em trabalhos sobre regulação de planos de saúde (ARAÚJO; SANTANA; NETO, 2015). As duas outras listas foram criadas segundo o método Máxima Relevância e Mínima Redundância (mRMR) utilizando-se a implementação disponibilizada pelos autores no site do projeto (MINIMUM. . . , 2016). O método mRMR possui dois modos de operação diferentes (modos MID e MIQ) e por isso é capaz de produzir duas listas de atributos possivelmente distintas. Esse método foi escolhido devido aos resultados positivos de sua aplicação em bases de dados com grandes quantidades de atributos (RATHORE; IFTIKHAR; HUSSAIN, 2014).

O resultado da aplicação do procedimento de seleção incremental de atributos em cada uma das 3 listas pode ser visto por meio da Figura 12 e da Tabela 10. Segundo a Figura 12, o subconjunto de atributos que apresentou a maior medida de acurácia durante

a avaliação está entre aqueles que contém entre 80 e 86 atributos ordenados segundo a Razão de Ganho de Informação. Mais precisamente, pela Tabela 10 observa-se que esse subconjunto é aquele formado pelos 83 primeiros atributos da lista correspondente ao método RGI (valor destacado na tabela).

A Figura 12 também mostra que o método Razão de Ganho de Informação nem sempre produziu os melhores subconjuntos de atributos. Se o objetivo fosse selecionar um subconjunto contendo entre 25 e 50 atributos, os mais bem avaliados seriam aqueles ordenados segundo o método Máxima Relevância Mínima Redundância no modo de operação MIQ.

Por conta desse resultado, os atributos selecionados para compor a base de dados Teresina Odontológico foram os 83 primeiros atributos contidos na lista criada com base no método Razão de Ganho de Informação.

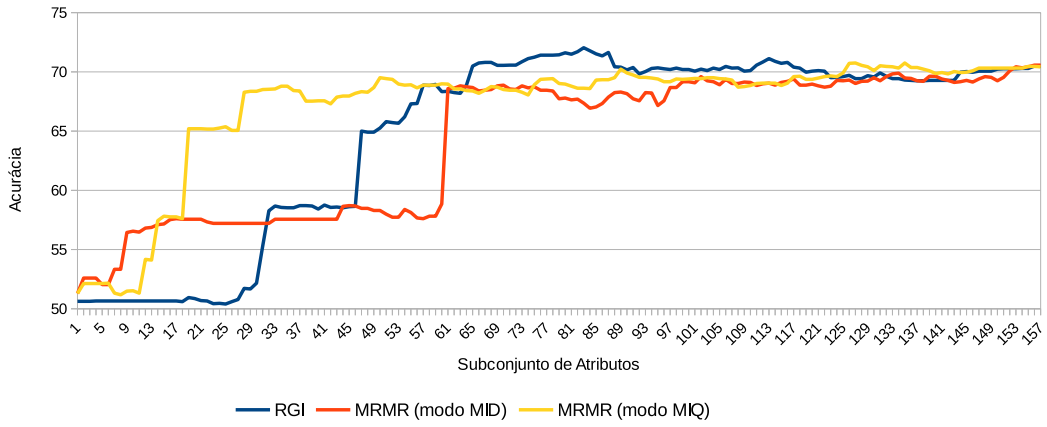

Figura 12 – Resultado da seleção de atributos segundo o procedimento de Seleção Incremental.

Tabela 10 – Parte dos valores de acurácia computados durante a seleção de atributos.

Subconjunto	Razão de Ganho de Informação	mRMR (modo MID)	mRMR (modo MIQ)
80	71.61	67.79	68.97
81	71.49	67.64	68.79
82	71.7	67.7	68.62
83	**72.04**	67.36	68.62
84	71.78	66.93	68.59
85	71.52	67.04	69.31
86	71.35	67.33	69.34