UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
PROGRAMA DE PÓS GRADUAÇÃO EM INFORMÁTICA

MARCOS ALEXANDRE PASTORI TERRIN

UTILIZANDO TÉCNICAS DE MINERAÇÃO DE DADOS PARA APOIAR A BUSCA ATIVA DE FAMÍLIAS EM SITUAÇÃO DE VULNERABILIDADE E RISCO SOCIAL

DISSERTAÇÃO DE MESTRADO

CORNÉLIO PROCÓPIO

2015

MARCOS ALEXANDRE PASTORI TERRIN

UTILIZANDO TÉCNICAS DE MINERAÇÃO DE DADOS PARA APOIAR A BUSCA ATIVA DE FAMÍLIAS EM SITUAÇÃO DE VULNERABILIDADE E RISCO SOCIAL

Dissertação de Mestrado apresentada ao Programa de Pós-Gradução em Informática da Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção do título de Mestre em Informática

Orientador: Prof. Dr. Carlos N. Silla Jr.

Coorientador: Prof. Dr. Pedro Henrique Bugatti.

CORNÉLIO PROCÓPIO

2015

Dados Internacionais de Catalogação na Publicação

T327 Terrin, Marcos Alexandre Pastori
Utilizando técnicas de mineração de dados para apoiar a busca ativa de famílias em situação de vulnerabilidade e risco social / Marcos Alexandre Pastori Terrin. – 2015.
133 f. : il. ; 30 cm

Orientador: Carlos Nascimento Silla Junior.
Coorientador: Pedro Henrique Bugatti.
Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Informática. Cornélio Procópio, 2015.
Bibliografia: p. 130-133.

1. Mineração de dados (Computação). 2. Assistência social. 3. Informática – Dissertações. I. Silla Junior, Carlos Nascimento, orient. II. Bugatti, Pedro Henrique, coorient. III. Universidade Tecnológica Federal do Paraná. Programa de Pós-graduação em Informática. IV. Título.

CDD (22. ed.) 004

Biblioteca da UTFPR, Câmpus Cornélio Procópio

Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Programa de Pós-Graduação em Informática

Título da Dissertação Nº 07:

"Utilizando Técnicas de Mineração de Dados para Apoiar a Busca Ativa de Famílias em Situação de Vulnerabilidade e Risco Social".

por

Marcos Alexandre Pastori Terrin

Orientador: **Prof. Dr. Carlos Nascimento Silla Junior**

Esta dissertação foi apresentada como requisito parcial à obtenção do grau de MESTRE EM INFORMÁTICA – Área de Concentração: Computação Aplicada, pelo Programa de Pós-Graduação em Informática – PPGI – da Universidade Tecnológica Federal do Paraná – UTFPR – Câmpus Cornélio Procópio, às 13h00 do dia 18 de agosto de 2015. O trabalho foi APROVADO pela Banca Examinadora, composta pelos professores:

Prof. Dr. Carlos Nascimento Silla Junior (Presidente)	Prof. Dr. Pedro Henrique Bugatti (Coorientador - UTFPR-CP)
Profa. Dra. Glaucia Maria Bressan (UTFPR-CP)	Prof. Dr. Paulo Rodrigo Cavalin (IBM Research Brazil)

Visto da coordenação:

Carlos Nascimento Silla Junior
Coordenador do Programa de Pós-Graduação em Informática
UTFPR Câmpus Cornélio Procópio

A Folha de Aprovação assinada encontra-se na Coordenação do Programa.

AGRADECIMENTOS

A minha esposa Lilian Maria Guedes Keller Terrin por ter me apoiado e compreendido o meu afastamento durante a fase de elaboração deste trabalho.

Aos meus pais Jovino Terrin e Lucia Pastori Terrin por terem se dedicado tanto em me dar amor, carinho, educação e caráter.

Especialmente aos meus professores e orientadores Prof. Dr. Carlos N. Silla Jr. e Prof. Dr. Pedro Henrique Bugatti que sempre acreditaram no meu potencial e me motivaram a estudar e a superar os desafios.

A todos os professores da UTFPR que tive contato durante as disciplinas do Programa de Pós-Graduação em Informática os quais considero como modelos de professores dedicados a ensinar.

A todos os meus amigos de turma que tornaram o ambiente de sala de aula um lugar agradável e divertido para aprender.

RESUMO

TERRIN, Marcos. UTILIZANDO TÉCNICAS DE MINERAÇÃO DE DADOS PARA APOIAR A BUSCA ATIVA DE FAMÍLIAS EM SITUAÇÃO DE VULNERABILIDADE E RISCO SOCIAL. 133 f. Dissertação de Mestrado – Programa de Pós Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

No âmbito da Assistência Social, existe a necessidade de se identificar as famílias em situação de vulnerabilidade e risco social, processo esse chamado de "Busca Ativa", para que as famílias nesta situação possam ser assistidas adequadamente. O Ministério do Desenvolvimento Social e Combate à Fome do Brasil orienta que seja realizado o cruzamento de bases de dados como forma de realizar a Busca Ativa, mas não disponibiliza nenhuma ferramenta para realização desse processo. Este trabalho busca identificar e aplicar técnicas de mineração de dados para apoiar a identificação das famílias em situação de vulnerabilidade e risco social. Os resultados obtidos em experimentos preliminares demonstraram que na maioria dos casos os modelos gerados preveem sempre a classe majoritária. Após realizar um balanceamento manual das classes removendo algumas amostras os experimentos foram repetidos e indicaram que os resultados estavam sendo diretamente afetados devido ao desbalanceamento das classes. Por esse motivo foram utilizados diversos métodos específicos para realizar o balanceamento das amostras a fim de que todas as classes possuíssem a mesma quantidade de amostras. Após realizar o balanceamento das amostras novos experimentos foram realizados. Durante a análise dos resultados foi observado que com as medidas padrões de avaliação de aprendizado de máquina não estava sendo possível identificar qual método havia obtido o melhor resultado. Em função disso um método de qualidade de *ranking* foi utilizado juntamente com a medida *Recall* para avaliar os resultados.

Palavras-chave: Busca ativa, Mineração de dados desbalanceados, Classificadores Bayesianos

ABSTRACT

TERRIN, Marcos. USING DATA MINING TECHNIQUES TO SUPPORT ACTIVE SEARCH FOR FAMILIES IN SITUATIONS OF SOCIAL RISK AND VULNERABILITY. 133 f. Dissertação de Mestrado – Programa de Pós Graduação em Informática, Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2015.

In the current Brazilian Government there is a Social Assistance policy that is highly concerned about helping families who might be at social risk and vulnerability. The process of identification of these families is known as "active search". The task of active search is defined in a document by the Brazilian Ministry of Social Development and Fight Against Hunger. This document provides the main guidelines about how to perform the active search. However, despite the task's importance, there are still no tool to help the social assistants with this task. This work aim to investigate the use of data mining techniques to identify the families in vulnerability and social risk situations. The results obtained in preliminary experiments showed that the classification models created always predict the majority class. After balancing manually the datasets by removing some examples the experiments were repeated and showed that the results were being directly influenced by the imbalanced data. Because of it was used a bunch of sampling methods to produce the same amount of examples in each class. After proceed with the sampling of the examples new experiments were proceeded. During the result's evaluation it was realized that the standard metrics used in machine learn were not being able to identify wich method obtained the best result. Due to this situation a ranking quality method was used combined with the Recall metric to evaluate the results.

Keywords: Active search, Unbalanced Data mining, Bayesian network classifiers.

LISTA DE FIGURAS

FIGURA 1 – Visibilidade das famílias vulneráveis para os sistemas 22
FIGURA 2 – Fluxograma de atendimento inicial das famílias nas unidades da rede .. 23
FIGURA 3 – Página principal do sistema IRSAS 24
FIGURA 4 – Fluxograma de preenchimento de Avaliação de Vulnerabilidade e Risco Social 26
FIGURA 5 – Primeira parte do questionário de avaliação de vulnerabilidade que avalia os dados de identificação do responsável 27
FIGURA 6 – Segunda parte do questionário de avaliação de vulnerabilidade que avalia as condições habitacionais da família 28
FIGURA 7 – Terceira parte do questionário de avaliação de vulnerabilidade que avalia o acesso ao conhecimento e escolarização 28
FIGURA 8 – Quarta parte do questionário de avaliação de vulnerabilidade que avalia as condições de saúde da família 29
FIGURA 9 – Quinta parte do questionário de avaliação de vulnerabilidade que avalia algumas condições gerais da família 29
FIGURA 10 – Sexta parte do questionário de avaliação de vulnerabilidade que avalia situações de acesso à profissionalização, trabalho e renda da família 30
FIGURA 11 – Resultado da avaliação de vulnerabilidade social onde é apresentado o índice e a classificação de vulnerabilidade 31
FIGURA 12 – Primeira parte do questionário de risco social que avalia situações de violação de direitos dos membros da família 32
FIGURA 13 – Segunda parte do questionário de risco social que avalia a situação de membros em cumprimento de medidas judiciais 33
FIGURA 14 – Resultado do questionário de risco social onde é apresentado o índice e a classificação de risco social 33
FIGURA 15 – Gráfico representando a quantidade de famílias que possuem avaliação de vulnerabilidade e risco social preenchidas no sistema IRSAS para a base de dados de Cascavel/PR 34
FIGURA 16 – Uma visão geral das etapas que compõem o processo de KDD 37
FIGURA 17 – Exemplo de vetores de características 40
FIGURA 18 – Visão geral da hierarquia do aprendizado indutivo 41
FIGURA 19 – Visão geral da construção de um modelo de classificação 42
FIGURA 20 – Espaço amostral para lançamento de dois dados, com evento E (resultado for "1") e evento F (resultado for "1") circulados em destaque 44
FIGURA 21 – Fórmula da equação do Teorema de Bayes para um espaço amostral binário com a origem de cada informação destacada 46
FIGURA 22 – Estrutura de um classificador *Naive Bayes* 49
FIGURA 23 – Estrutura de um classificador TAN 52
FIGURA 24 – Exemplos de estruturas de rede com quatro atributos preditivos para os seguintes classificadores de rede Bayesiana: NB, TAN e KDB1, AODE ... 54
FIGURA 25 – Ilustração de uma tabela de exemplo de matriz de confusão 55
FIGURA 26 – Matriz de confusão para duas classes 55

FIGURA 27 – Várias amostras da classe majoritária (negativas) com algumas amostras da classe minoritária (positivas) esparramadas (a) conjunto de dados balanceado com agrupamentos bem definidos (b) 62
FIGURA 28 – Uma ilustração de como são criados os pontos sintéticos no Algoritmo *SMOTE* 66
FIGURA 29 – Balanciando um conjunto de dados: conjunto de dados original (a); conjunto de dados após *oversampling* (b); Identificação dos Tomek links (c); e remoção dos ruídos e amostras de fronteira (d) 67
FIGURA 30 – (a) Distribuição original do conjunto de dados Circle. (b) As amostras de fronteira da classe minoritária (quadrados sólidos). (c) As amostras de fronteira sintéticas (quadrados vazios) 71
FIGURA 31 – Relação de tabelas originárias dos atributos utilizados e seus relacionamentos 83
FIGURA 32 – Parte do conjunto de dados extraído do sistema IRSAS com dados faltantes 84
FIGURA 33 – Parte do conjunto de dados extraído do sistema IRSAS após transformação de dados faltantes 85
FIGURA 34 – Parte do arquivo ARFF gerado para ser utilizado no WEKA 86
FIGURA 35 – Matrizes de confusão do experimento 1 89
FIGURA 36 – Matrizes de confusão do experimento 2 90
FIGURA 37 – Matrizes de confusão do experimento 3 91
FIGURA 38 – Matrizes de confusão do experimento 4 92
FIGURA 39 – Matrizes de confusão do experimento 5 93
FIGURA 40 – Matrizes de confusão do experimento 6 94
FIGURA 41 – Protocolo experimental 99
FIGURA 42 – Média das matrizes de confusão dos resultados do classificador Naive Bayes com o conjunto de dados M_04 da Database 02 com método de balanceamento SMOTE_ENN 108
FIGURA 43 – Média das matrizes de confusão dos resultados do classificador Naive Bayes com o conjunto de dados D_04 da Database 02 com método de balanceamento SMOTE_ENN 109
FIGURA 44 – Média das matrizes de confusão dos resultados do classificador AODE com o conjunto de dados M_04 da Database 02 com método de balanceamento SPIDER 110
FIGURA 45 – Gráfico Recall vs DCG Dataset M_04 Database 01 116
FIGURA 46 – Gráfico Recall vs DCG Dataset D_04 Database 01 117
FIGURA 47 – Gráfico Recall vs DCG Dataset M_04 Database 02 118
FIGURA 48 – Gráfico Recall vs DCG Dataset D_04 Database 02 119

LISTA DE TABELAS

TABELA 1 – Classificação de Vulnerabilidade 25
TABELA 2 – Valores DCG 59
TABELA 3 – Exemplo da geração de amostra sintética pelo SMOTE 66
TABELA 4 – A definição de ruído, fronteira e região segura no *Borderline-SMOTE* .. 69
TABELA 5 – Atributos selecionados 83
TABELA 6 – Relação de atributos descritos na Tabela 5 utilizados nos experimentos preliminares 87
TABELA 7 – Distribuição das amostras utilizadas nos experimentos 1 ao 4 88
TABELA 8 – Resultados do experimento 1 utilizando validação cruzada fator 10 88
TABELA 9 – Resultados do experimento 2 utilizando validação cruzada fator 10 89
TABELA 10 – Resultado do experimento 3 utilizando validação cruzada fator 10 91
TABELA 11 – Resultado do experimento 4 utilizando validação cruzada fator 10 91
TABELA 12 – Distribuição das amostras utilizadas nos experimentos 5 e 6 92
TABELA 13 – Resultados do experimento 5 utilizando validação cruzada fator 10 93
TABELA 14 – Resultados do experimento 6 utilizando validação cruzada fator 10 94
TABELA 15 – Distribuição das amostras nas duas bases de dados de acordo com a classe de vulnerabilidade 97
TABELA 16 – Parâmetros de configuração dos métodos de balanceamento na ferramenta KEEL 102
TABELA 17 – Resultados Dataset M_04 Database 01 103
TABELA 18 – Resultados Dataset D_04 Database 01 104
TABELA 19 – Resultados Dataset M_04 Database 02 105
TABELA 20 – Resultados Dataset D_04 Database 02 106
TABELA 21 – Resultado do classificador Naive Bayes para o Dataset M_04 da Database 02 com o método de balanceamento SMOTE_ENN 107
TABELA 22 – Resultado do classificador Naive Bayes para o Dataset D_04 da Database 02 com o método de balanceamento SMOTE_ENN 109
TABELA 23 – Resultado do classificador AODE para o Dataset M_04 da Database 02 com o método de balanceamento SPIDER 110
TABELA 24 – Exemplo DCG 112
TABELA 25 – Exemplo DCG 2 113
TABELA 26 – Resultados parciais do classificador AODE para o Dataset M_04 da Database 02 113
TABELA 27 – Resultados parciais do classificador AODE para o Dataset D_04 da Database 02 115
TABELA 28 – Melhores resultados Database 01 124
TABELA 29 – Melhores resultados Database 02 124

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
AODE	*Averaged One-Dependence Estimators*
ARFF	*Attribute-Relation File Format*
BPC	Benefício de Prestação Continuada
CadÚnico	Cadastro Único
CG	*Cumulative Gain*
CRAS	Centro de Referência da Assistência Social
CREAS	Centro de Referência Especializado de Assistência Social
CTAN	*Construct-TAN*
DAG	*Directed Acyclic Graph*
DCG	*Discounted Cumulative Gain*
ENN	*Edited Nearest Neighbor*
GPL	*General Public License*
IBGE	Instituto Brasileiro de Geografia e Estatística
IRSAS	Informatização da Rede de Serviços da Assistência Social
KDBC	*K-Dependence Bayesian Classifier*
KDD	*Knowledge Discovery in Databases*
KEEL	*Knowledge Extraction based on Evolutionary Learning*
KNN	*K-Nearest Neighbor*
LBR	*Lazy Bayesian Rules*
MDS	Ministério do Desenvolvimento Social e Combate à Fome
NB	*Naive Bayes*
ONG	Organização Não Governamental
PAIF	Proteção e Atendimento Integral à Família
PETI	Programa de Erradicação do Trabalho Infantil
RBC	Raciocínio Baseado em Casos
RN	Redes Neurais
ROS	*Random over-sampling*
SEMISH	Seminário Integrado de Software e Hardware
SMOTE	*Synthetic Minority Over-sampling TEchnique*
SPIDER	*Selective Preprocessing of Imbalanced Data*
SPODEs	*Superparent One-Dependence Estimators*
SP-TAN	*Super-Parent TAN*
SQL	*Structured Query Language*
TAN	*Tree Augmented Naive Bayes*
UBS	Unidade Básica de Saúde
WEKA	*Waikato Environment for Knowledge Analysis*

LISTA DE SÍMBOLOS

E,F,G	Eventos aleatórios
P(E)	Probabilidade de E
P(F)	Probabilidade de F
P(G)	Probabilidade de G
N	Quantidade de ocorrências de um conjunto de eventos
M	Quantidade de ocorrências de um evento específico
$P(E \text{ ou } F)$, $P(E \cup F)$	Probabilidade da união de E e F
$P(E \text{ e } F)$, $P(E \cap F)$	Probabilidade da intersecção dos eventos E e F
$P(\overline{E})$	Probabilidade do complemento do evento E
$P(E\|F)$	Probabilidade condicional de E dado F
$P(F\|E)$	Probabilidade condicional de F dado E
$\{E1, E2, E3, ..., En\}$	Conjunto de eventos simples
S	Espaço de resultados elementares, espaço amostral
$\sum$	Representação da soma de um grande número de termos, somatório
$\overline{E}$	Complemento do evento E
A_i	Atributo de uma instância
C	Conjunto de rótulos ou classes
a_i	Valor do atributo de uma instância
t	Representação de uma instância
c	Representação do valor do rótulo ou classe
$c(t)$	Representação da classe ou rótulo que t pertence
$argmax$	Argumento que maximiza uma função
$\in$	Pertença a conjunto
$\prod$	Produto
L	Dimensão da um lado da matriz
V_p	Número de exemplos positivos da classe que foram previsto corretamente
F_n	Número de exemplos negativos da classe que foram previsto incorretamente
F_p	Número de exemplos positivos da classe que foram classificados incorretamente
V_n	Número de exemplos negativos da classe que foram previsto corretamente
n	Tamanho de uma amostra
r	Número de partições de uma amostra
p	Posição de um documento em uma lista ordenada
rel_i	Relevância de um determinado documento na posição i
$\log_2 i$	Função logarítmica de um número i na base 2
D_i	Representação de um documento em uma lista ordenada
k	Número de itens vizinhos considerados por um método
x_i	Ponto do espaço amostral selecionado pelo algoritmo de *oversampling*

r_i	Pontos sintéticos criados pelo algoritmo de *oversampling*
$d(E_i,\ E_j)$	Distância entre duas instâncias
$Nmin$	Número de amostras da classe minortiária
NS	Quantidade percentual de amostras a serem criadas pelo SMOTE
$Nmaj$	Número de amostras da classe majoritária
CTR	Conjunto de dados de treinamento
CMI	Classe minoritária
CMA	Classe majoritária
cmi_i	Amostra da classe minoritária
cma_i	Amostra da classe majoritária
$dnum$	Número de amostras no conjunto *DANGER*
s	Inteiro entre 1 e k
m	Número de vizinhos mais próximos de uma instância
m'	Número de vizinhos da classe majoritária mais próximos de uma instância
v	Vizinho mais próximo selecionado de *cmi*
f	Instância sintética
sl_ratio	*Safe Level Ratio*
$numattrs$	Número de atributos
dif	Diferença entre os valores dos atributos v e *cmi*
gap	Fração aleatória de *dif*
$cmi[i]$, $v[i]$, $f[i]$	Valores numéricos de um determinado atributo da amostra na i^{a} posição
sl_{cmi}, sl_v	Safe Level Ratio de *cmi*, e v respectivamente
∞	Infinito
ns	Número exemplos sintéticos criados
ampl	Opçâo de amplificação dos algoritmos SPIDER e SPIDER2
o	Rótulo dado a uma instância pelos algoritmos SPIDER e SPIDER2
DS	Conjunto de dados
z	Número de cópias geradas pela função de amplificação do algoritmo SPIDER 2

SUMÁRIO

1 INTRODUÇÃO 14
1.1 CONTEXTUALIZAÇÃO 14
1.2 MOTIVAÇÃO E JUSTIFICATIVA 15
1.3 OBJETIVOS 16
1.3.1 Objetivo Geral 16
1.3.2 Objetivos Específicos 16
1.4 ORGANIZAÇÃO DO TEXTO 17
2 PROBLEMA 19
2.1 ESTADO DA ARTE PARA O PROBLEMA 20
2.2 ESTADO DA ARTE PARA AVALIAÇÃO DE VULNERABILIDADE E RISCO SOCIAL 25
2.2.1 Avaliação de Vulnerabilidade do IRSAS 26
2.2.2 Avaliação de Risco Social do IRSAS 31
2.3 CONSIDERAÇÕES FINAIS 33
3 REFERENCIAL TEÓRICO 36
3.1 *KNOWLEDGE DISCOVERY IN DATABASES* 36
3.2 APRENDIZADO DE MÁQUINA 39
3.3 PROBABILIDADE CONDICIONAL E TEOREMA DE BAYES 43
3.4 REDES BAYESIANAS 48
3.5 CLASSIFICADORES BAYESIANOS 48
3.6 MÉTODOS PARA A AVALIAÇÃO DE MODELOS 54
3.6.1 Matriz de confusão 54
3.6.2 *Cross-Validation* 57
3.6.3 *Leave-one-out* 57
3.7 DCG (*DISCOUNTED CUMULATIVE GAIN*) 57
3.8 WEKA 60
4 MINERAÇÃO DE DADOS DESBALANCEADOS 61
4.1 ROS - RANDOM OVER-SAMPLING 64
4.2 SMOTE: SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE 64
4.3 SMOTE + TOMEK LINKS 66
4.4 SMOTE + ENN 68
4.5 BORDERLINE-SMOTE 68
4.6 SAFE-LEVEL-SMOTE 72
4.7 SPIDER 75
4.8 SPIDER2 78
4.9 KEEL 80
5 METODOLOGIA 81
5.1 SELEÇÃO 82
5.2 PRÉ-PROCESSAMENTO 84
5.3 FORMATAÇÃO 85
5.4 MINERAÇÃO DE DADOS 85

5.5 INTERPRETAÇÃO/AVALIAÇÃO 86
6 EXPERIMENTOS PRELIMINARES 87
6.1 CONFIGURAÇÃO EXPERIMENTAL 87
6.2 EXPERIMENTO 1 - UTILIZANDO TODOS OS ATRIBUTOS DA FAMÍLIA 88
6.3 EXPERIMENTO 2 - UTILIZANDO APENAS OS ATRIBUTOS "CHAVES" DEFINIDOS POR UM ESPECIALISTA DO DOMÍNIO 89
6.4 EXPERIMENTO 3 - USANDO APENAS OS ATRIBUTOS DO FORMULÁRIO DE AVALIAÇÃO DE VULNERABILIDADE 90
6.5 EXPERIMENTO 4 - UTILIZANDO OS ATRIBUTOS DO FORMULÁRIO DE AVALIAÇÃO EM CONJUNTO COM OS ATRIBUTOS DO EXPERIMENTO 1 91
6.6 EXPERIMENTO 5 - USANDO APENAS OS ATRIBUTOS DO FORMULÁRIO DE AVALIAÇÃO DE VULNERABILIDADE COM CLASSES BALANCEADAS 93
6.7 EXPERIMENTO 6 - UTILIZANDO OS ATRIBUTOS DO FORMULÁRIO DE AVALIAÇÃO DE VULNERABILIDADE EM CONJUNTO COM OS ATRIBUTOS DO EXPERIMENTO 1 COM CLASSES BALANCEADAS 93
6.8 ANÁLISE DOS EXPERIMENTOS PRELIMINARES 94
7 EXPERIMENTOS 96
7.1 CONFIGURAÇÃO EXPERIMENTAL 96
7.2 PROTOCOLO EXPERIMENTAL 97
7.3 AVALIAÇÃO DOS EXPERIMENTOS 107
7.3.1 Avaliação através da Precisão 107
7.3.2 Avaliação através do *Recall* 108
7.3.3 Avaliação através do F-measure 110
7.3.4 Proposta de uma nova forma de avaliação 111
7.3.5 Avaliação através do DCG 113
7.3.6 Uma nova forma de avaliação através do *Recall* e DCG 114
7.3.7 Avaliação dos resultados 120
7.4 ANÁLISE DOS EXPERIMENTOS 125
8 CONSIDERAÇÕES FINAIS 127
8.1 TRABALHOS FUTUROS 129
REFERÊNCIAS 130

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Os Centros de Referência da Assistência Social (CRAS) são unidades de atendimento espalhadas pelos diversos territórios de uma cidade, que surgiram através da descentralização da secretaria de assistência social, de modo a aproximar o contato dessa instituição com a população. O Ministério do Desenvolvimento Social e Combate à Fome do Brasil (MDS) divulgou em 2009 em sua cartilha chamada "Orientações Técnicas Centro de Referência de Assistência Social - CRAS" (BRASIL, 2009), uma série de orientações para coordenar as atividades realizadas pelos CRAS. Essa cartilha identifica a Busca Ativa, como a busca intencional realizada pela equipe de referência do CRAS, visando conhecer o território e as famílias que ali residem. A cartilha ainda ressalta que a identificação e o conhecimento das situações de vulnerabilidade e risco social devem ser utilizados como fonte para o planejamento municipal, para a definição de serviços e para a ação preventiva nos territórios dos CRAS.

A principal contribuição deste trabalho é identificar e aplicar técnicas de mineração de dados para auxiliar a identificação das famílias em situação de vulnerabilidade e risco social, através de informações coletadas pela equipe da assistência social no município de Cascavel/PR, no intuito de facilitar e apoiar o processo da Busca Ativa das famílias.

O sistema de Informatização da Rede de Serviços da Assistência Social (IRSAS), apresentado no Capítulo 2, possui uma ferramenta não automatizada para geração do índice de vulnerabilidade e risco social das famílias. Esse índice é gerado após o preenchimento de um formulário de avaliação de vulnerabilidade que, além das perguntas específicas, utiliza dados do cadastro da família para realizar o cálculo do índice de vulnerabilidade. Devido ao processo não ser automatizado e demandar a coleta de várias informações sobre a família, a geração da avaliação de vulnerabilidade é feita com apenas 0,8% das famílias que já estão em acompanhamento pelos CRAS. Desse modo, atualmente a avaliação de vulnerabilidade não auxilia no processo de Busca Ativa, pois não abrange a geração de classificação para as famílias que não estão sendo ativamente assistidas pela rede de serviços da assistência social.

Este é um trabalho pioneiro que trata de um problema real que impacta diretamente nas famílias carentes que estão em situação de vulnerabilidade e risco social necessitando serem assistidas pelo Estado. Através das pesquisas realizadas durante a realização deste trabalho não foi encontrado nenhum outro trabalho similar cujo foco fosse a utilização de técnicas computacionais para automatizar o processo de identificação dessas famílias.

Além de classificar corretamente as famílias em um dos rótulos de vulnerabilidade, é importante que o algoritmo de classificação utilizado possa também aferir o grau de probabilidade da família estar em situação de vulnerabilidade. Uma vez que o intuito final da solução é identificar quais famílias precisam ser prioritariamente assistidas. Desse modo, ao invés de obter as famílias rotuladas em uma das classes de vulnerabilidade, será possível obter uma estimativa de probabilidade da família ser vulnerável e consequentemente uma lista das famílias em situação de maior vulnerabilidade. Através dessa estimativa será possível fornecer uma lista ordenada de prioridade de atendimento com base na probabilidade de vulnerabilidade, apoiando o processo da Busca Ativa. Essa lista ordenada das famílias torna-se necessária, uma vez que o município pode não possuir capacidade suficiente para atender todas as famílias vulneráveis simultaneamente.

Pretende-se posteriormente utilizar as técnicas de mineração de dados experimentadas neste trabalho que deram resultados positivos para implementar uma ferramenta automatizada no sistema IRSAS para apoiar o processo da Busca Ativa das famílias em situação de vulnerabilidade e risco social. Com esta ferramenta os municípios poderão identificar famílias vulneráveis que ainda não estejam sendo acompanhadas pela equipe da assistência social.

1.2 MOTIVAÇÃO E JUSTIFICATIVA

Do ponto de vista social, a Busca Ativa das famílias em situação de vulnerabilidade social é muito importante para que o Estado possa assistir as famílias mais necessitadas que por algum motivo não procuram de forma espontânea os serviços da assistência social.

Atualmente não existe nenhuma ferramenta específica para realização da Busca Ativa das famílias para os municípios utilizarem. Dessa forma, o presente trabalho busca identificar e aplicar técnicas de mineração de dados para viabilizar a construção de uma ferramenta que auxilie no processo da Busca Ativa dessas famílias.

O autor já possui uma ferramenta utilizada por alguns municípios brasileiros para cadastramento e controle das pessoas atendidas pelos serviços da assistência social. Pretende-se com esse estudo, incrementar essa ferramenta para apoiar o processo da Busca Ativa das

famílias em situação de vulnerabilidade e risco social através da classificação automatizada do grau de vulnerabilidade e risco social das famílias.

1.3 OBJETIVOS

Diante da recomendação do MDS, para que os municípios realizem a Busca Ativa através da utilização de informações existentes, o presente trabalho pretende estudar e aplicar técnicas de mineração de dados para apoiar o processo da Busca Ativa utilizando as informações reais coletadas através do sistema de Informatização da Rede de Serviços da Assistência Social (IRSAS), que será apresentado no Capítulo 2 e que atualmente é utilizado por três municípios, sendo eles Londrina/PR, Cascavel/PR e Mogi das Cruzes/SP.

Pretende-se investigar e aplicar técnicas de mineração de dados nos dados reais coletados através do sistema IRSAS e analisar os resultados dos experimentos a fim de averiguar se os algoritmos são adequados para obtenção de resultados e se os mesmos são capazes de indicar uma classificação de vulnerabilidade confiável para as famílias.

Como resultado, pretende-se obter uma técnica de classificação satisfatória para auxiliar os municípios a realizarem a classificação de vulnerabilidade das famílias, auxiliando no processo da Busca Ativa das famílias que estiverem em situação de vulnerabilidade e risco social, através da classificação obtida pelo modelo de predição.

1.3.1 OBJETIVO GERAL

- Utilizar técnicas de Mineração de Dados para apoiar o processo da Busca Ativa das famílias em situação de vulnerabilidade e risco social.

1.3.2 OBJETIVOS ESPECÍFICOS

- Utilizar o processo do KDD (Knowledge Discovery in Databases) para a descoberta de conhecimento nas bases de dados de famílias em situação de vulnerabilidade e risco social;
- Aplicar técnicas de classificação nos dados existentes a fim de obter uma classificação de vulnerabilidade e risco social para apoiar o processo da Busca Ativa;
- Comparar os resultados obtidos, com as classificações de vulnerabilidades existentes e avaliar os resultados para mensurar o nível de acerto das técnicas automatizadas;

1.4 ORGANIZAÇÃO DO TEXTO

Estre trabalho está divido em 8 capítulos. Além do capítulo inicial de introdução do trabalho que esclarece a contextualização, motivação e justificativa, objetivos e organização do documento, os demais capítulos são os seguintes:

- **Problema:** neste capítulo é apresentado de forma detalhada qual o problema central deste trabalho. Em função do problema estar relacionado com a área da Assistência Social, esse capítulo se faz necessário para apresentar o estado da arte do problema e esclarecer de forma detalhada as situações reais e termos que são de suma importância para compreensão do trabalho;

- **Referencial Teórico:** neste capítulo são apresentadas as tecnologias e estudos utilizados para realização do trabalho. São elas: O processo de descoberta de conhecimento (KDD), Aprendizado de Máquina, Redes Bayesianas, Métodos para a avaliação de modelos, Métodos para avaliação de qualidade de *ranking* e por fim a ferramenta WEKA (HALL et al., 2009), utilizada para realização dos experimentos;

- **Mineração de Dados Desbalanceados:** neste capítulo são apresentados os desafios da mineração de dados em conjuntos de dados desbalanceados, os principais problemas, as técnicas existentes para lidar com este problema, os métodos de balanceamento das amostras nos conjuntos de dados utilizados no trabalho, e por fim a ferramenta KEEL (ALCALA-FDEZ et al., 2009), (ALCALA-FDEZ et al., 2011), utilizada para realização do balanceamento das amostras nos conjuntos de dados de treinamento;

- **Metodologia:** neste capítulo são apresentadas como cada etapa do Processo do KDD foi realizada. Sendo elas: Seleção de atributos, pré-processamento, formatação, mineração de dados e por fim a interpretação e avaliação dos resultados;

- **Experimentos Preliminares:** neste capítulo são apresentados os experimentos preliminares realizados neste trabalho. Para cada experimento realizado é apresentado: a combinação dos dados utilizados, os algoritmos testados, os resultados obtidos, e a interpretação e análise dos resultados;

- **Experimentos:** neste capítulo são apresentados os experimentos com métodos de balanceamento realizados neste trabalho. Para cada experimento realizado é apresentado: a combinação dos dados utilizados, os algoritmos testados, os resultados obtidos, e a interpretação e análise dos resultados;

- **Considerações Finais:** neste capítulo são apresentadas as considerações finais do trabalho realizado, as avaliações feitas e os trabalhos futuros.

2 PROBLEMA

O Plano Brasil sem Miséria foi lançado em junho de 2011 pelo governo federal. O objetivo do plano é elevar a renda e as condições de bem-estar da população. Para isso, as famílias extremamente pobres que ainda não eram atendidas, deveriam ser localizadas e incluídas de forma integrada nos diversos programas sociais do governo federal (Ex.: Programa Bolsa Família), de acordo com as suas necessidades. Esse plano é direcionado aos brasileiros que vivem em lares cuja renda familiar é de até R$ 70 por pessoa. De acordo com o Censo 2010 do Instituto Brasileiro de Geografia e Estatística (IBGE), estão nesta situação 16,2 milhões de brasileiros. Para isso, o Plano Brasil sem Miséria desenvolveu uma nova estratégia chamada "Busca Ativa". O Ministério do Desenvolvimento Social e Combate à Fome (BRASIL, 2009) descreve a Busca Ativa como a procura intencional realizada pela equipe de referência do Centro de Referência da Assistência Social (CRAS) tendo como objetivo identificar as famílias em situação de vulnerabilidade e risco social para ampliar o conhecimento e a compreensão da realidade social, auxiliar no planejamento municipal, para definição de serviços socioassistenciais a serem ofertados em cada território, para a ação preventiva no território dos CRAS e principalmente levar o Estado até as famílias menos favorecidas inserindo-as nas políticas públicas adequadas.

O MDS em BRASIL (2009) indica que a Busca Ativa deve-se apoiar em informações disponíveis, oriundas do sistema do Cadastro Único (CadÚnico), que contém informações a respeito das famílias que participam do Programa Bolsa Família. Algumas dessas informações são de domínio público mas as maioria são de acesso restrito aos técnicos da assistência social do município. O MDS ressalta que as informações devem ser incorporadas no processo de trabalho, utilizando-as para definir ações estratégicas, urgentes, preventivas e de rotina.

No documento de orientações técnicas para os CRAS em BRASIL (2009), o MDS sugere algumas estratégias para realizar a Busca Ativa, sendo elas:

- Deslocamento da equipe de referência para conhecimento do território;
- Contatos com atores sociais locais (líderes comunitários, associações de bairro, etc.);

- Obtenção de informações e dados provenientes de outros serviços socioassistenciais e setoriais;
- Campanhas de divulgação, distribuição de panfletos, colagem de cartazes e utilização de carros de som.

Outra estratégia de realização da Busca Ativa é a utilização de dados das famílias do território de atuação do CRAS proveniente do Cadastro Único, de Programas Sociais e de algumas listagens disponíveis para o município, como por exemplo:

- Dos beneficiários do Benefício de Prestação Continuada (BPC);
- Dos beneficiários do Programa de Erradicação do Trabalho Infantil (PETI);
- Dos beneficiários do Programa Bolsa Família;
- Dos beneficiários do Programa Bolsa Família em descumprimento de condicionalidades.

O MDS também orienta que seja realizado o cruzamento de bases de dados, extraídas de alguns sistemas estaduais e federais e eventuais dados coletados pelo município, como forma de realizar a Busca Ativa, e garante autonomia para os municípios utilizarem a melhor forma disponível, mas não disponibiliza nenhuma ferramenta para viabilizar esse processo (BRASIL, 2013).

Apesar das atualizações e inclusões constantes desses sistemas, existem milhares de famílias em extrema pobreza que deveriam, mas ainda não estão incluídas em políticas e programas sociais adequados. Essa situação implica na invisibilidade dessas famílias diante dos municípios, estados e do governo federal.

2.1 ESTADO DA ARTE PARA O PROBLEMA

O MDS indica diversas formas de realizar a Busca Ativa. Particularmente, as técnicas de Busca Ativa que dizem respeito ao cruzamento de informações das famílias são baseadas em informações oriundas principalmente do sistema CadÚnico e outros sistemas que acompanham famílias inseridas em programas sociais. Entretanto, ao realizar a Busca Ativa utilizando os dados desses sistemas, o universo de famílias fica limitado àquelas que já foram ou estão inseridas em algum programa social. Ou seja, utilizando apenas essas fontes de informação a Busca Ativa não será capaz de identificar uma grande quantidade de famílias que necessitam ser

inseridas nas políticas públicas de assistência social, e que até o momento são desconhecidas perante o Estado. Desse modo, a Busca Ativa, através do cruzamento de informações indicadas pelo MDS, não é suficiente para identificar as famílias que estão em situação de vulnerabilidade e que permanecem invisíveis perante a assistência social.

A principal limitação do CadÚnico é que ele só possui informações referentes às famílias cadastradas no programa Bolsa Família. Uma alternativa para realização da Busca Ativa pode ser a utilização de informações coletadas pelo sistema de Informatização da Rede de Serviços da Assistência Social (IRSAS). Ao contrário do sistema CadÚnico, utilizado apenas nos CRAS para fins de inclusão dos cidadãos no programa Bolsa Família, o IRSAS é utilizado por todas as unidades da rede de serviços da assistência social de um determinado município. Em função disto, o IRSAS é a porta de entrada de muitas famílias que são atendidas por serviços dos municípios de Cascavel/PR, Londrina/PR e Mogi das Cruzes/SP e que não necessariamente estão inseridas no CadÚnico.

O IRSAS é um sistema de informação que funciona de forma integrada em diversas entidades do município, por exemplo, CRAS; Centro de Referência Especializado de Assistência Social (CREAS); Escolas Municipais e Estaduais; Unidades Básicas de Saúde; Conselhos Tutelares; Albergues; Associação de Pais e Amigos dos Excepcionais; Entidades Sociais da Rede Governamental e não Governamental; Unidades de Acolhimento Institucional; entre outras. Atualmente no município de Cascavel/PR existem 319 unidades utilizando o sistema.

Um exemplo de situação onde o IRSAS pode ser utilizado por uma família que não faz parte do CadÚnico é, por exemplo, caso uma mãe procure uma organização não governamental (ONG) que atenda pessoas com deficiência auditiva para inserir seu filho em algum programa, a ONG irá cadastrar essa família no IRSAS. Caso essa família não seja encaminhada para o programa Bolsa Família, essa família não será inserida na base de dados do CadÚnico e continuará invisível para o sistema federal. Desse modo o universo de famílias mantidas na base de dados do IRSAS é maior que as mantidas na base de dados do CadÚnico, conforme ilustrado na Figura 1. Pressupõe-se que as famílias inseridas no CadÚnico já são conhecidas pela assistência social pois precisam ser acompanhadas pela equipe técnica dos CRAS para que sejam inseridas no programa Bolsa Família.

Tendo em vista que o principal objetivo da Busca Ativa é encontrar famílias em situação de vulnerabilidade e risco social que ainda não estejam sendo assistidas devidamente pelos órgãos competentes, a realização da Busca Ativa na base de dados do IRSAS possibilita encontrar mais famílias nessa situação do que quando realizada apenas com os dados do CadÚnico.

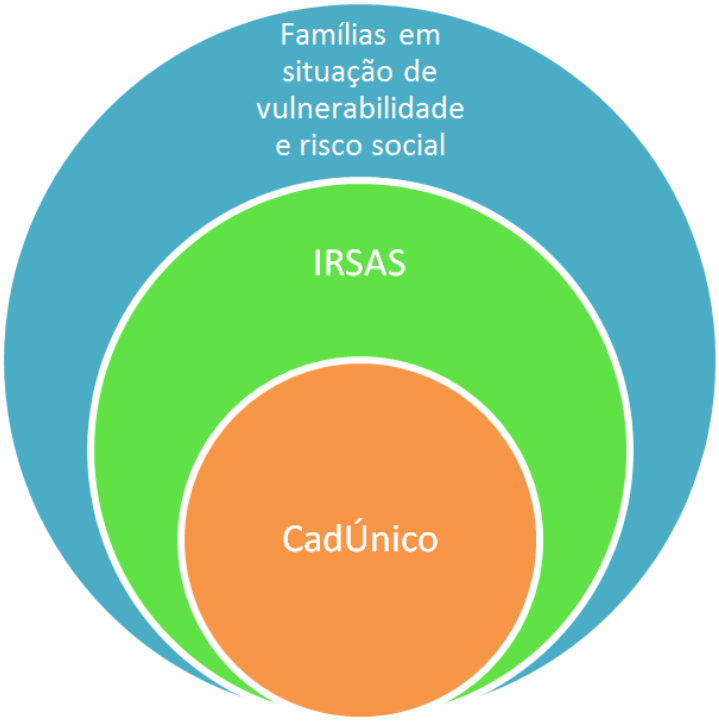


Figura 1: Visibilidade das famílias vulneráveis para os sistemas
Fonte: Autoria própria.

Sempre que uma pessoa procura o CRAS ou qualquer outra unidade da rede de serviços da assistência social para solicitar ajuda, como por exemplo, para inclusão no programa Bolsa Família, o técnico responsável pelo atendimento irá consultar se essa pessoa e sua família já estão cadastradas no IRSAS. Caso não estejam, o primeiro passo será providenciar as informações para realizar o cadastramento dessa família no sistema. Uma vez cadastrados, os dados passam a ser compartilhados com toda a rede de serviços que utiliza o IRSAS. Caso essa família procure atendimento em outro serviço da rede, o técnico irá consultar e constatar que essa família já possui cadastro. Nesse caso a segunda unidade irá apenas efetuar atualizações nos dados cadastrais quando necessário antes de prosseguir com o atendimento.

A Figura 2 apresenta o fluxograma de atendimento inicial das famílias nas unidades da rede de assistência social que utilizam o IRSAS.

A Figura 3 apresenta a página principal do IRSAS implantado em 2011 no município de Cascavel/PR onde atualmente existem 132.675 pessoas cadastradas que agrupadas somam um total de 43.016 famílias.

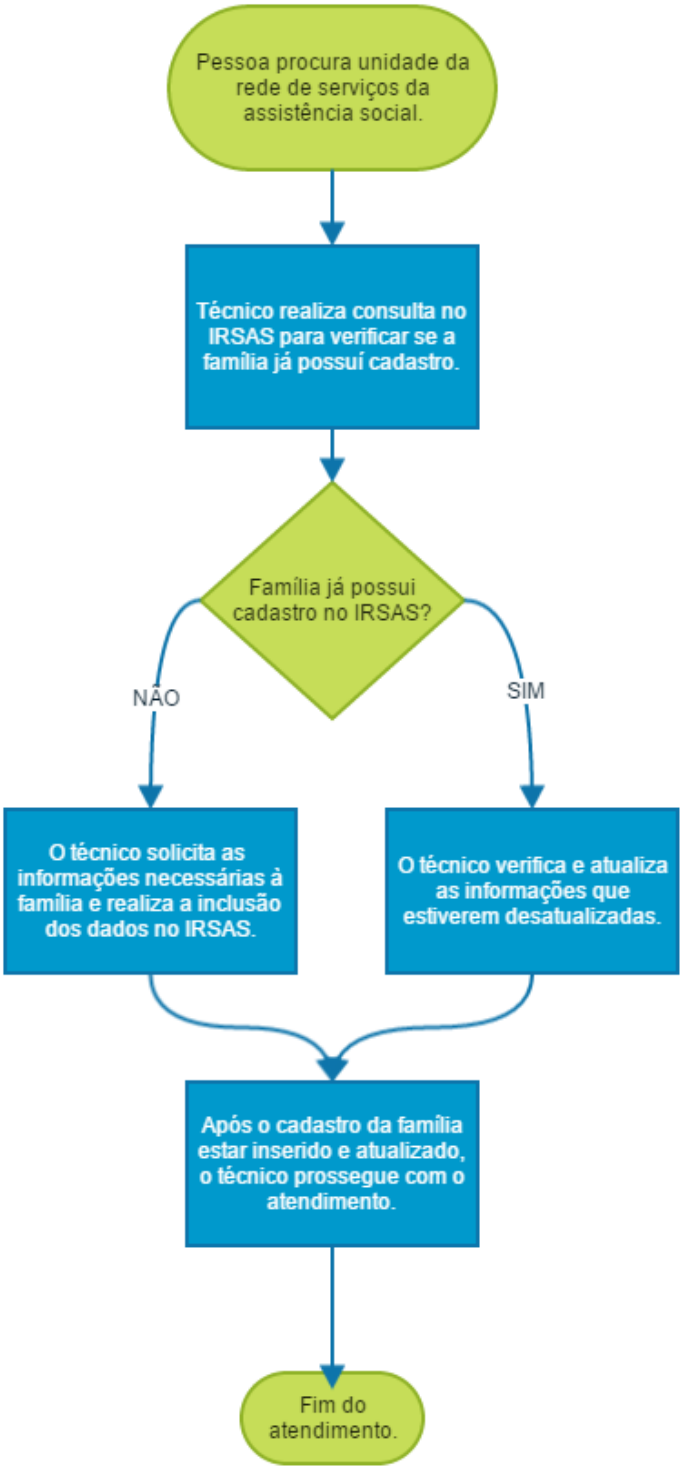


Figura 2: Fluxograma de atendimento inicial das famílias nas unidades da rede

Fonte: Autoria própria.

Figura 3: Página principal do sistema IRSAS

Fonte: Autoria própria.

Não existe hoje nenhum sistema disponível para realizar a Busca Ativa das famílias em situação de vulnerabilidade e Risco Social. Contudo, existe no sistema IRSAS um formulário chamado de "Avaliação de Vulnerabilidade e Risco Social", apresentado na Seção 2.2, que apoia a avaliação da situação de vulnerabilidade e risco social das famílias em acompanhamento pelos técnicos dos CRAS.

2.2 ESTADO DA ARTE PARA AVALIAÇÃO DE VULNERABILIDADE E RISCO SOCIAL

Ao contrário do cadastro da família que é feito para 100% das pessoas atendidas nos serviços da rede de assistência social com o IRSAS, a Avaliação de Vulnerabilidade e Risco Social é feita apenas para pessoas que estejam em acompanhamento direto pela equipe técnica do CRAS e estejam prioritariamente inseridas no Serviço de Proteção e Atendimento Integral à Família (PAIF).

O PAIF é um trabalho de caráter continuado que visa fortalecer a função de proteção das famílias, prevenindo a ruptura de laços, promovendo o acesso e usufruto de direitos e contribuindo para a melhoria da qualidade de vida, tendo como público famílias em situação de vulnerabilidade social.

Porém apesar da Avaliação de Vulnerabilidade e Risco Social ser um instrumento para identificar o nível de vulnerabilidade e risco social das famílias, ele não é obrigatório para que as famílias sejam acompanhadas pelos técnicos do CRAS. Devido a limitações de tempo, recursos humanos e logísticos (transporte, condução) para levar os técnicos até a residência das famílias para coleta dos dados, o formulário de avaliação de vulnerabilidade e risco social do IRSAS acaba não sendo amplamente utilizado pelos técnicos.

A Figura 4 apresenta o fluxograma de preenchimento do formulário de Avaliação de Vulnerabilidade e Risco Social.

O formulário de avaliação de vulnerabilidade e risco social foi desenvolvido para medir o nível de vulnerabilidade e risco social das famílias através de diversas questões elaboradas pela equipe técnica da secretaria municipal de assistência social de Cascavel/PR. Todas as questões possuem uma pontuação vinculada a suas respostas de modo que ao final do preenchimento desse formulário é gerado um índice de vulnerabilidade e risco social da família. Esse índice varia de 0 (zero) a 100 (cem) e gera uma classificação categórica que pode ser de “Baixa”, “Média” ou “Alta” vulnerabilidade conforme apresentado na Tabela 1.

Tabela 1: Classificação de Vulnerabilidade

Pontuação	Classificação
<= 15	Baixa Vulnerabilidade
>= 16 e <= 30	Média Vulnerabilidade
>= 31	Alta Vulnerabilidade

Fonte: Autoria própria.

O formulário é dividido em duas seções, a primeira seção é referente à avaliação de

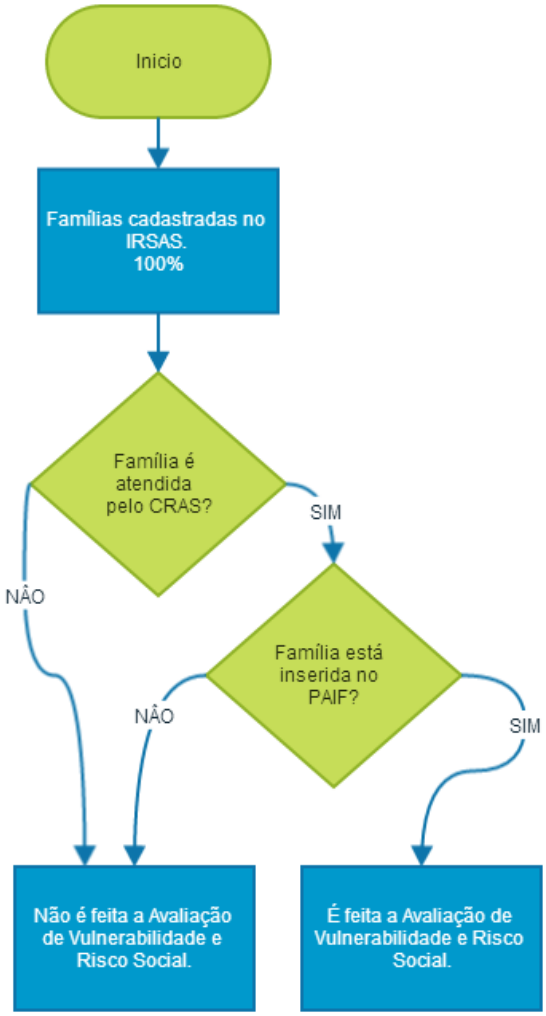


Figura 4: Fluxograma de preenchimento de Avaliação de Vulnerabilidade e Risco Social
Fonte: Autoria própria.

vulnerabilidade e a segunda seção é referente ao risco social. Estas seções do formulário são apresentadas em detalhes nas Seções 2.2.1 e 2.2.2.

2.2.1 AVALIAÇÃO DE VULNERABILIDADE DO IRSAS

A primeira seção do formulário de avaliação de vulnerabilidade e risco social do IRSAS corresponde a avaliação de vulnerabilidade, e é composta por seis partes. A primeira parte (Figura 5) avalia os dados de identificação do responsável (raça, sexo, idade, deficiência) e também identifica se algum outro membro da família possui algum tipo de deficiência.

A segunda parte (Figura 6) avalia as condições habitacionais da família (tipo de logradouro, número de cômodos, condição de moradia, etc.).

Figura 5: Primeira parte do questionário de avaliação de vulnerabilidade que avalia os dados de identificação do responsável

Fonte: Autoria própria.

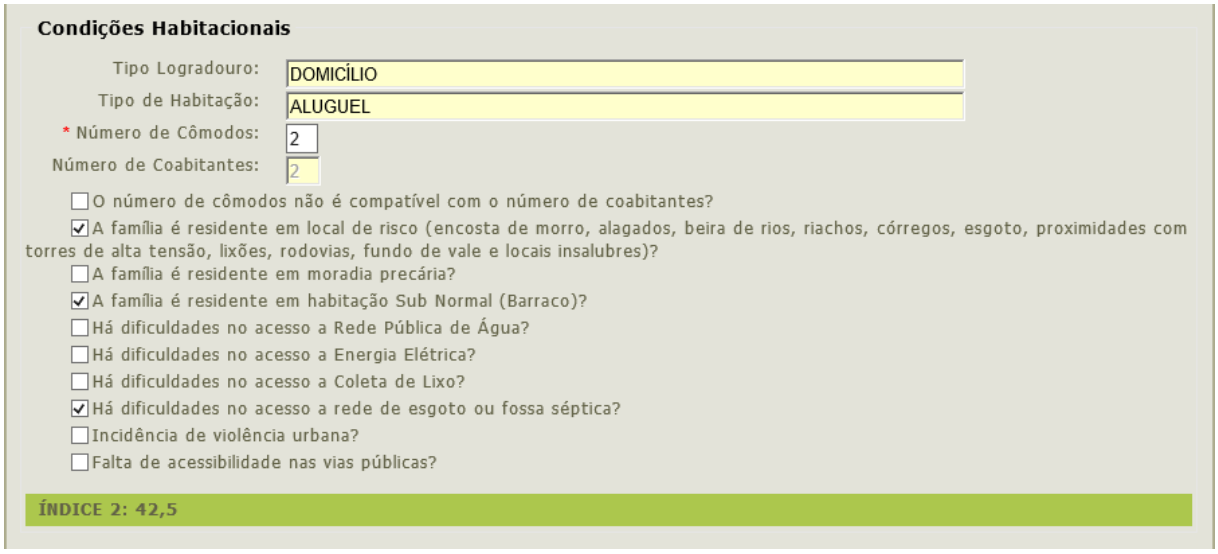


Figura 6: Segunda parte do questionário de avaliação de vulnerabilidade que avalia as condições habitacionais da família

Fonte: Autoria própria.

A terceira parte (Figura 7) avalia o acesso ao conhecimento, escolarização do responsável e se existem membros da família em defasagem escolar.

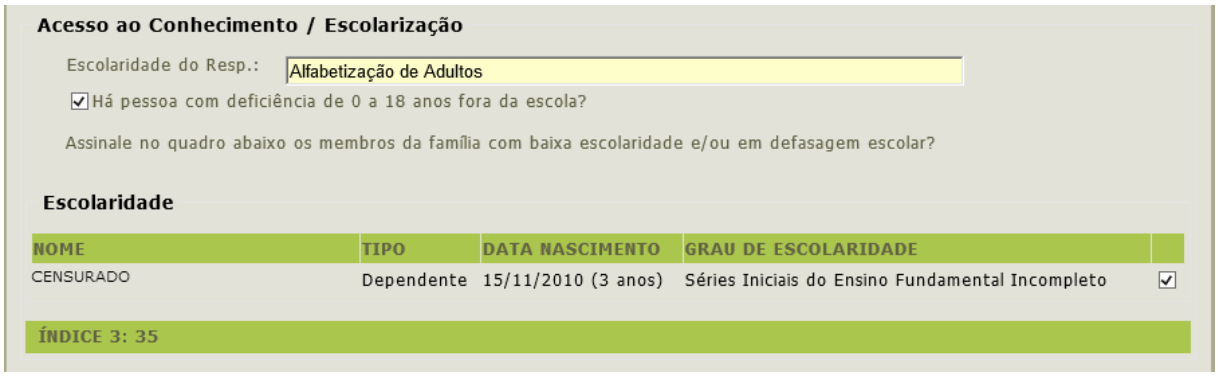


Figura 7: Terceira parte do questionário de avaliação de vulnerabilidade que avalia o acesso ao conhecimento e escolarização

Fonte: Autoria própria.

A quarta parte (Figura 8) avalia as condições de saúde da família (doenças crônicas, pessoas acamadas, gravidez na adolescência, etc.).

A quinta parte (Figura 9) avalia algumas condições gerais da família (responsável monoparental, se crianças de 0 a 12 anos ficam sozinhas no domicílio, descumprimento de condicionalidades dos programas sociais, etc.).

A sexta parte (Figura 10) avalia situações de acesso à profissionalização, trabalho e

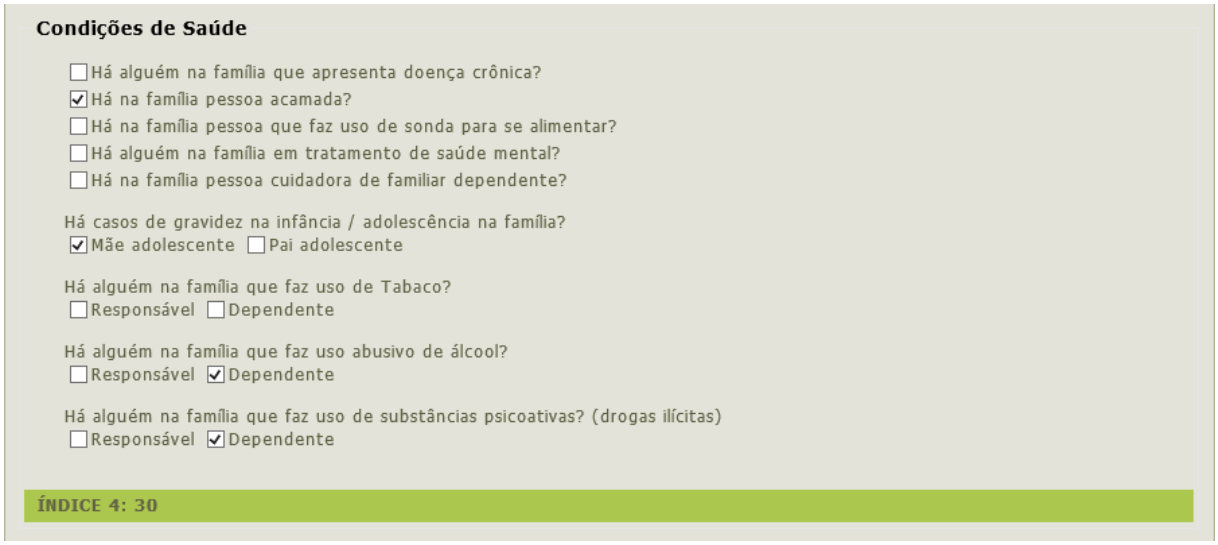


Figura 8: Quarta parte do questionário de avaliação de vulnerabilidade que avalia as condições de saúde da família

Fonte: Autoria própria.

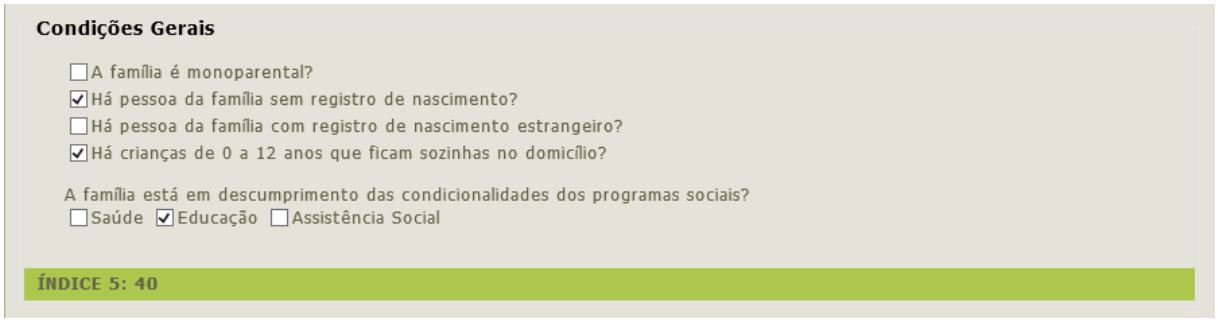
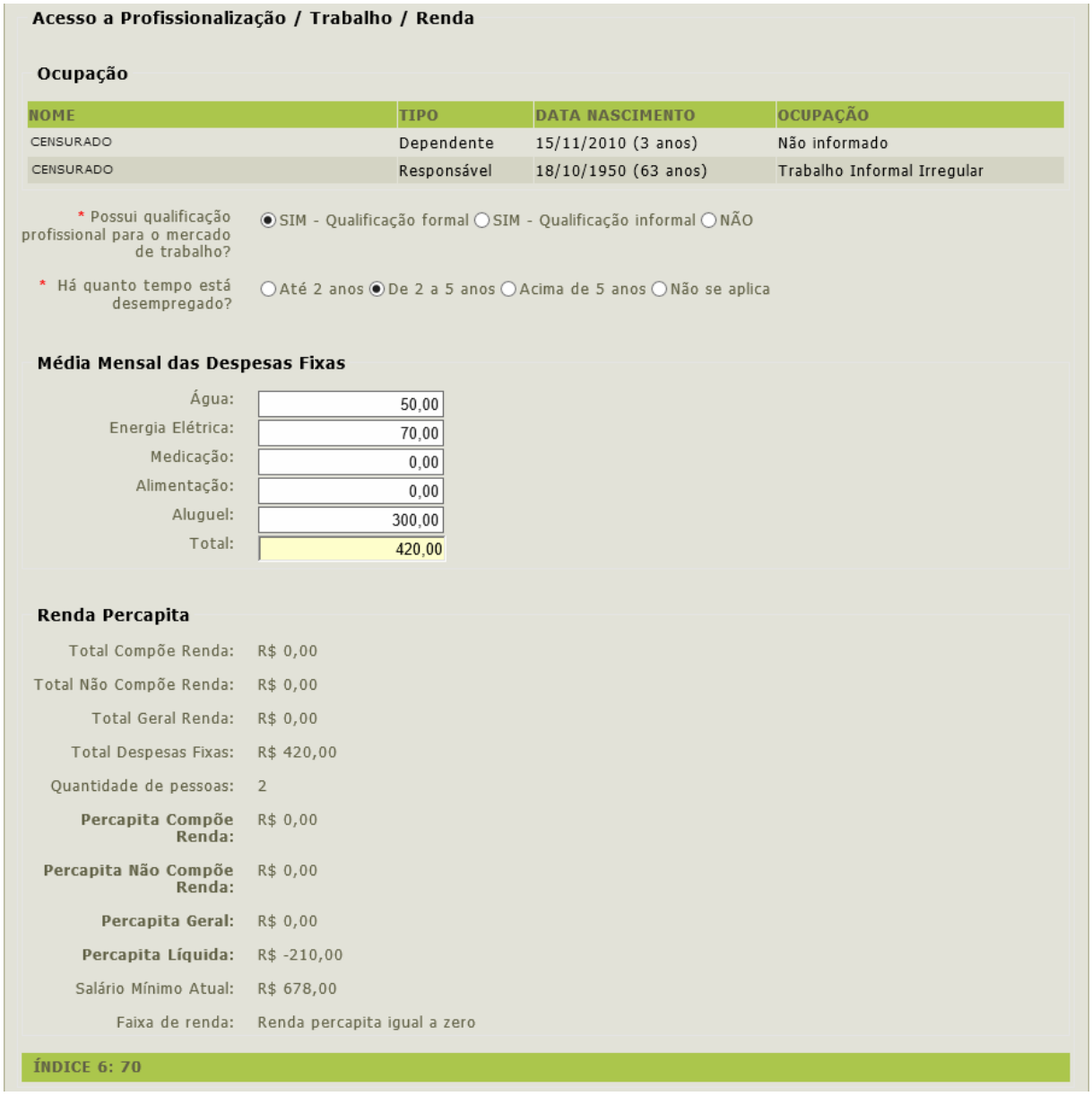


Figura 9: Quinta parte do questionário de avaliação de vulnerabilidade que avalia algumas condições gerais da família

Fonte: Autoria própria.

renda da família (ocupação dos membros, qualificação para o mercado de trabalho, média das despesas fixas e rendas da família).



Acesso a Profissionalização / Trabalho / Renda

Ocupação

NOME	TIPO	DATA NASCIMENTO	OCUPAÇÃO
CENSURADO	Dependente	15/11/2010 (3 anos)	Não informado
CENSURADO	Responsável	18/10/1950 (63 anos)	Trabalho Informal Irregular

* Possui qualificação profissional para o mercado de trabalho? ◉ SIM - Qualificação formal ○ SIM - Qualificação informal ○ NÃO

* Há quanto tempo está desempregado? ○ Até 2 anos ◉ De 2 a 5 anos ○ Acima de 5 anos ○ Não se aplica

Média Mensal das Despesas Fixas

Água:	50,00
Energia Elétrica:	70,00
Medicação:	0,00
Alimentação:	0,00
Aluguel:	300,00
Total:	420,00

Renda Percapita

Total Compõe Renda:	R$ 0,00
Total Não Compõe Renda:	R$ 0,00
Total Geral Renda:	R$ 0,00
Total Despesas Fixas:	R$ 420,00
Quantidade de pessoas:	2
Percapita Compõe Renda:	R$ 0,00
Percapita Não Compõe Renda:	R$ 0,00
Percapita Geral:	R$ 0,00
Percapita Líquida:	R$ -210,00
Salário Mínimo Atual:	R$ 678,00
Faixa de renda:	Renda percapita igual a zero

ÍNDICE 6: 70

Figura 10: Sexta parte do questionário de avaliação de vulnerabilidade que avalia situações de acesso à profissionalização, trabalho e renda da família

Fonte: Autoria própria.

A Figura 11 apresenta o resultado obtido através das respostas das questões das partes 1 a 6 do formulário de avaliação de vulnerabilidade. Nessa parte é apresentada a pontuação obtida e a classificação (alta, média ou baixa vulnerabilidade).

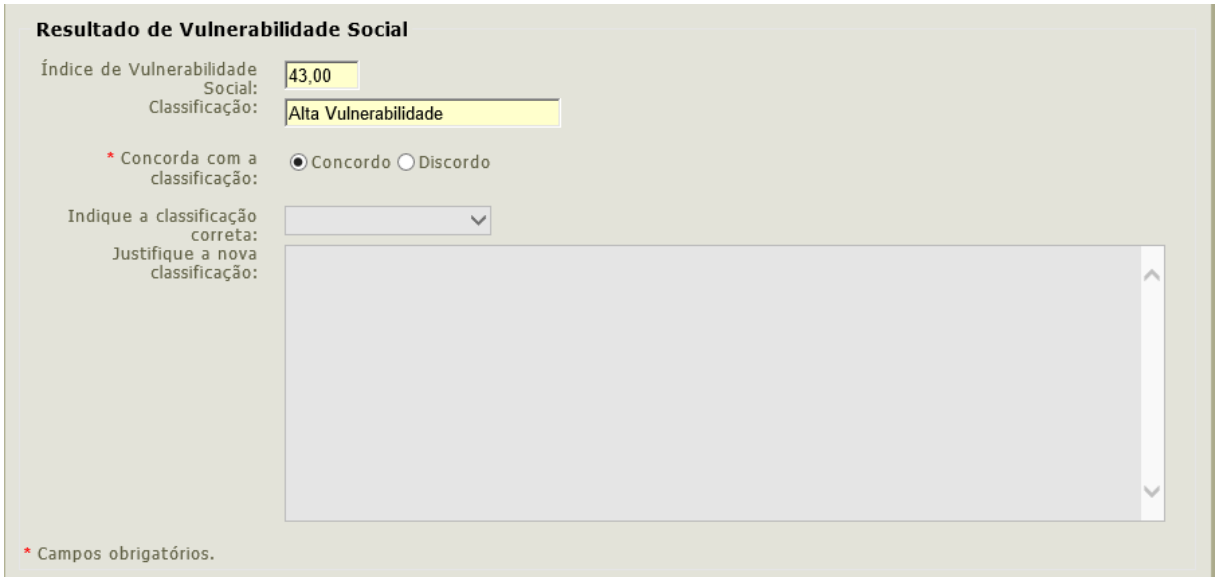


Figura 11: Resultado da avaliação de vulnerabilidade social onde é apresentado o índice e a classificação de vulnerabilidade

Fonte: Autoria própria.

2.2.2 AVALIAÇÃO DE RISCO SOCIAL DO IRSAS

A segunda seção do formulário (avaliação de risco social) é composta por duas partes. A primeira parte (Figura 12) avalia situações de violação de direitos dos membros da família (casos de violências).

A segunda parte (Figura 13) avalia a situação de membros em cumprimento de medidas judiciais (medidas socioeducativas e situações de acolhimento institucional).

A Figura 14 apresenta os resultados obtidos através das respostas das questões das partes 1 e 2 do formulário de risco social. Nessa parte é apresentada a pontuação obtida e a classificação (alto, médio ou baixo risco social).

Figura 12: Primeira parte do questionário de risco social que avalia situações de violação de direitos dos membros da família

Fonte: Autoria própria.

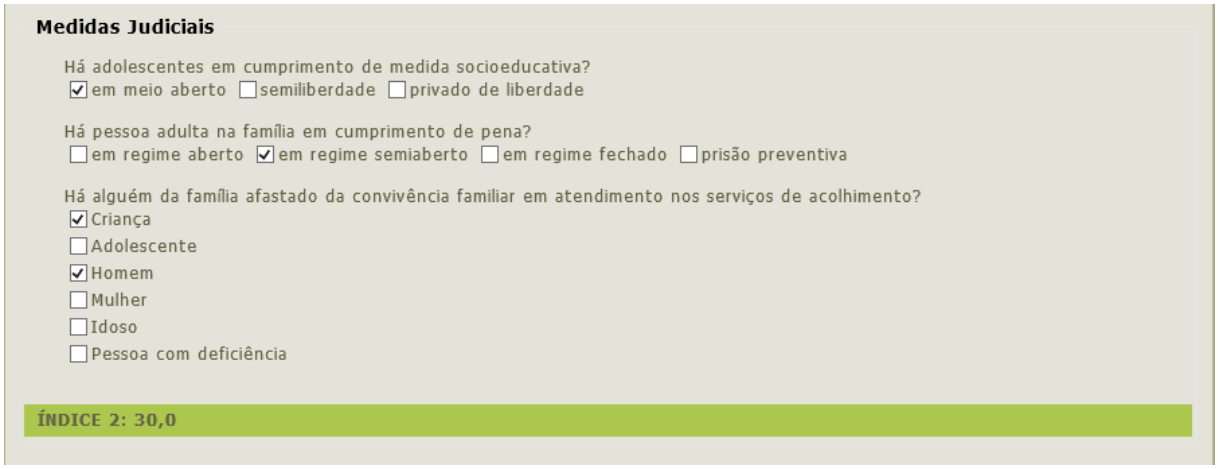


Figura 13: Segunda parte do questionário de risco social que avalia a situação de membros em cumprimento de medidas judiciais

Fonte: Autoria própria.

Figura 14: Resultado do questionário de risco social onde é apresentado o índice e a classificação de risco social

Fonte: Autoria própria.

2.3 CONSIDERAÇÕES FINAIS

Atualmente a avaliação de vulnerabilidade e risco social do IRSAS tem sido utilizada em cenários como:

- identificar, dentre as famílias em acompanhamento, quais delas necessitam de maior atenção e ajuda naquele determinado momento;
- acompanhamento da evolução da situação de vulnerabilidade e risco social das famílias em acompanhamento;
- critério de classificação para participação em programas municipais (Ex.: Vale Gás), sendo que as famílias que obtiverem o maior índice de vulnerabilidade possuem prioridade na participação dos programas;
- forma de identificar a vulnerabilidade dos territórios da cidade através do cruzamento dos resultados das avaliações de vulnerabilidade com os endereços das famílias.

Porém é importante ressaltar que apenas uma pequena parte das famílias cadastradas no IRSAS (348 famílias de 43.016 para o município de Cascavel/PR) possui o formulário de Avaliação de Vulnerabilidade e Risco Social preenchido, já que para isso é necessário que a família esteja em acompanhamento direto pelos técnicos do CRAS e também que seja realizada uma visita presencial até a residência de cada família para a coleta de informações e confirmação das respostas cadastradas no IRSAS pelo técnico do CRAS. Na base do município de Cascavel/PR o volume de famílias com avaliação de vulnerabilidade e risco social preenchida é de apenas 0,8% conforme apresentado na Figura 15.

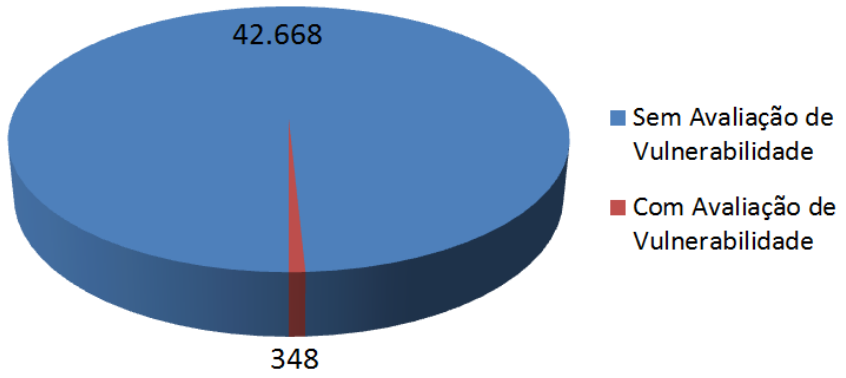


Figura 15: Gráfico representando a quantidade de famílias que possuem avaliação de vulnerabilidade e risco social preenchidas no sistema IRSAS para a base de dados de Cascavel/PR

Fonte: Autoria própria.

Desse modo, é possível observar que a classificação de vulnerabilidade e risco social está disponível apenas para uma pequena parte das famílias cadastradas. Apesar de ser um instrumento importante para o acompanhamento da situação de vulnerabilidade do público alvo

da assistência social, a Avaliação de Vulnerabilidade e Risco Social é utilizada de forma específica para um pequeno grupo de famílias atendidas deixando de lado uma grande quantidade de famílias que possivelmente estejam em situação de maior vulnerabilidade e risco social do que aquelas que já estão sendo acompanhadas.

Com o atual cenário em vista, o problema principal a ser estudado é como realizar a classificação de vulnerabilidade e risco social para 100% das famílias cadastradas no sistema utilizando apenas as informações já existentes no cadastro das famílias sem que seja necessário realizar o preenchimento do formulário de Avaliação de Vulnerabilidade e Risco Social existente no IRSAS.

Com isso, será possível identificar o nível de vulnerabilidade e risco social de todas as famílias e consequentemente encontrar aquelas com maior índice de vulnerabilidade e que por algum motivo não estão sendo assistidas devidamente pela assistência social. Por exemplo, caso uma família tenha um filho deficiente auditivo inserido em algum programa de uma organização não governamental (ONG), essa família estará sendo atendida por essa unidade, porém se essa família estiver em situação de alta vulnerabilidade e risco social ela deverá ser acompanhada ativamente pelas demais unidades da rede de serviço de assistência social. Nesse exemplo a família é atendida por uma das unidades, mas mesmo assim continua a não ser atendida devidamente pelas demais unidades da rede de serviços da assistência social.

Ou seja, desenvolver uma técnica capaz de apoiar o processo da Busca Ativa das famílias em situação de vulnerabilidade e risco social com base nas informações comuns existentes em todas as famílias cadastradas no sistema sem que seja necessário realizar a visita e o preenchimento do questionário.

3 REFERENCIAL TEÓRICO

Devido à redução dos custos de equipamentos de armazenamento, quantidades enormes de informações são armazenadas diariamente. Essas informações dificilmente são analisadas e transformadas em conhecimento estratégico para as empresas. Um dos objetivos da mineração de dados é a descoberta de conhecimento através de técnicas computacionais, que são capazes de explorar um grande conjunto de dados evidenciando padrões e auxiliando na descoberta de conhecimento.

Witten et al. (2011) destacam que estamos sobrecarregados com informações. Os computadores tornam muito fácil armazenar dados que anteriormente seriam descartados. Na medida em que o volume de informação aumenta, inexoravelmente, a proporção que as pessoas a compreendem diminui de forma alarmante. Repousa escondida em toda essa informação, conhecimento potencialmente útil que raramente será explicitado e utilizado de forma vantajosa.

Nesse capítulo é destacado como é o processo de descoberta de conhecimento em bases de dados, conhecido em inglês como *Knowledge Discovery in Databases* (KDD), e qual a sua relação com a mineração de dados.

3.1 *KNOWLEDGE DISCOVERY IN DATABASES*

O termo KDD refere-se a um processo criado para transformar dados brutos armazenados em um banco de dados em conhecimento. Segundo Fayyad et al. (1996) o KDD é um processo não trivial de identificação de padrões válidos, novos, potencialmente úteis e, por fim, compreensíveis, encontrados nos dados.

Comumente o termo mineração de dados é confundido com o próprio KDD, mas o processo de descoberta de conhecimento é dividido em várias etapas sendo a mineração de dados uma delas.

Cada uma das etapas do KDD tem um objetivo específico e o conjunto de todas elas resulta na geração de conhecimento através da exploração de informações de uma base de dados.

O processo do KDD pode envolver várias iterações e podem conter repetições entre duas quaisquer das etapas do processo. Uma visão geral do fluxo básico das etapas do KDD é apresentada na Figura 16.

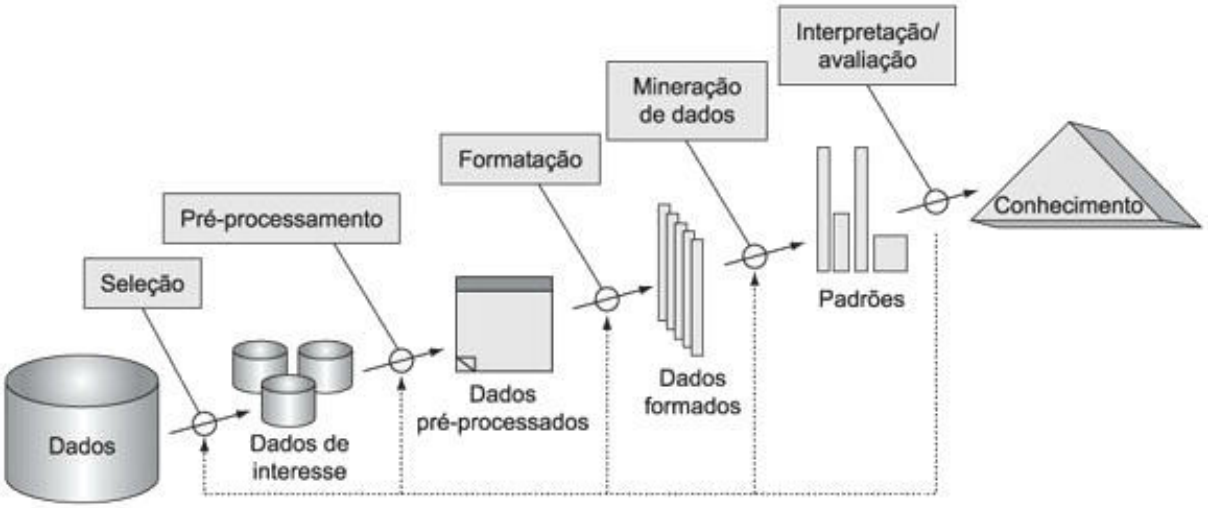


Figura 16: Uma visão geral das etapas que compõem o processo de KDD
Fonte: (FAYYAD et al., 1996)

As etapas do KDD apresentadas na Figura 16 podem ser descritas brevemente segundo Fayyad et al. (1996) como sendo:

- **Seleção:**

  Seleção ou segmentação dos dados apropriados para a análise de acordo com algum critério relacionado ao objetivo a ser alcançado. Desta maneira, subconjuntos dos dados podem ser determinados. No contexto desse trabalho, por exemplo, poderiam ser selecionados os dados de todas as pessoas com renda familiar inferior a um quarto de salário mínimo;

- **Pré-processamento:**

  Eliminação de ruídos e erros existentes, estabelecimento de procedimentos para verificação da falta de dados, estabelecimento de convenções para nomeação e outros passos para a construção de uma base de dados consistente. Por exemplo, a verificação e substituição de valores faltantes nos atributos por valores que não afetarão de forma negativa os resultados alcançados;

- **Formatação:**

Redução dos dados através da busca por atributos que mais representem a informação de acordo com o objetivo a ser alcançado. Nessa etapa, o número de atributos efetivos pode ser reduzido. Um exemplo dessa etapa é a seleção de atributos onde são utilizadas técnicas para descobrir quais dos atributos selecionados possuem mais informação a respeito das classes no caso de um problema de aprendizado supervisionado;

- **Mineração de Dados:**

  Aplicação dos algoritmos para descoberta de padrões nos dados envolve a seleção de métodos/técnicas e modelos que melhor se enquadram no cumprimento do objetivo a ser alcançado. Essa etapa frequentemente envolve a repetição de aplicações iterativas de métodos de mineração de dados. No contexto desse trabalho, por exemplo, foram utilizados algoritmos de classificação supervisionados;

- **Interpretação/Avaliação:**

  Interpretação dos padrões encontrados, possibilitando o retorno para qualquer um dos passos anteriores para uma nova iteração. Essa etapa pode envolver a visualização dos padrões e modelos extraídos ou a visualização das informações dado um modelo extraído. Podem ser utilizadas diversas técnicas para visualização e interpretação dos resultados obtidos visando sempre à obtenção da informação.

Segundo Fayyad et al. (1996), após a realização destas etapas, o conhecimento descoberto pode ser incorporado a outro sistema para ações futuras, ou simplesmente documentado e reportado para outras partes interessadas. Esse processo também inclui a verificação e potencial resolução de conflitos com conhecimentos adquiridos ou extraídos anteriormente.

Dentro do processo do KDD, o objetivo da descoberta de conhecimento é definido pela intenção do usuário do sistema. Fayyad et al. (1996) categorizam os tipos de objetivos como:

- **Verificação**, onde o sistema é limitado a verificar uma hipótese definida pelo usuário;
- **Descoberta**, onde o sistema procura de forma automatizada por padrões.

O objetivo de **descoberta** ainda é subdividido em dois tipos:

- **Predição**, onde o sistema procura por padrões para prever o comportamento de alguma entidade;
- **Descrição**, onde o sistema procura por padrões que são representados de uma forma compreensiva para o usuário.

Contudo, as barreiras entre **predição** e **descrição** não são tão lineares. Alguns modelos de predição podem ser descritivos ao ponto de serem compreensíveis e vice versa. Essa distinção é útil para entendimento do objetivo geral da descoberta. Os objetivos da predição e descrição podem ser alcançados utilizando uma variedade de métodos de aprendizado de máquina.

3.2 APRENDIZADO DE MÁQUINA

Existe na literatura uma sobreposição dos termos Aprendizado de Máquina, Reconhecimento de Padrões e Mineração de Dados. Esses termos são frequentemente utilizados para se referir à mesma área da inteligência artificial. Segundo MONARD e BARANAUSKAS (2003), o Aprendizado de Máquina (AM) é uma área da inteligência artificial cujo objetivo é o desenvolvimento de técnicas computacionais sobre o aprendizado e também a construção de sistemas inteligentes capazes de obter conhecimento de forma automática.

Dentro da área de Aprendizado de Máquina existem diversos paradigmas para realização do aprendizado. De acordo com MONARD e BARANAUSKAS (2003), a maioria das abordagens existentes podem ser enquadradas em um destes paradigmas:

- **Simbólico:** os sistemas de aprendizado simbólico constroem representações simbólicas de um conceito através da análise de exemplos. Caracteristicamente as representações simbólicas são expressas em forma de alguma expressão lógica, árvore de decisão, regras ou rede semântica;
- **Estatístico:** a ideia geral desses sistemas consiste em utilizar modelos estatísticos para encontrar uma boa aproximação do conceito induzido. Um dos métodos estatísticos que mais se destaca é o de aprendizado Bayesiano, que utiliza um modelo probabilístico baseado no conhecimento prévio do problema;
- **Baseado em Exemplo:** esse tipo de sistema utiliza exemplos anteriores de classe conhecida para classificar um novo exemplo com base em sua similaridade. Um dos algoritmos mais conhecidos desse tipo é o Nearest Neighbours e Raciocínio Baseado em Casos (RBC);
- **Conexionista:** modelos conexionistas são aqueles que possuem unidades altamente interconectadas, como por exemplo, as Redes Neurais (RN);
- **Genético:** esse modelo consiste de uma população de elementos de classificação que competem para fazer a predição. Os elementos com melhor desempenho são mantidos

e os fracos são descartados. A ideia central é que os elementos fortes sejam mantidos e possam se proliferar produzindo variações de si mesmos.

Existe uma grande quantidade de algoritmos de aprendizado de máquina e cada um pode apresentar desempenho diferente de acordo com o problema. Desse modo, não existe um único algoritmo capaz de obter o melhor resultado em todas as situações, sendo importante compreender o poder e as limitações de cada um. Este conceito tratado por Wolpert e Macready (1997) ficou amplamente conhecido na literatura como *"No Free Lunch Theorem"*.

De acordo com MONARD e BARANAUSKAS (2003), a indução é a forma de inferência lógica que possibilita obter conclusões genéricas através de um conjunto de exemplos e o aprendizado indutivo pode ser dividido entre supervisionado e não-supervisionado. Geralmente os exemplos utilizados no aprendizado indutivo são compostos por um vetor de características contendo os atributos de cada exemplo. A Figura 17 apresenta um exemplo dos vetores de características de 3 exemplos, cada vetor contém 3 atributos e o rótulo da classe a qual cada exemplo pertence.

Id.	Atrib 1	Atrib 2	Atrib 3	Classe
1	Sim	Grande	125K	Não
2	Não	Médio	100K	Não
3	Não	Pequeno	70K	Não

Figura 17: Exemplo de vetores de características
Fonte: Adaptado de (TAN et al., 2009)

- No **aprendizado supervisionado** o algoritmo, também chamado de indutor, recebe um conjunto de exemplos de treinamento. Geralmente esses exemplos são compostos por um vetor de características e o rótulo da classe associada. Nesse caso, o objetivo do algoritmo indutor é construir um classificador, com base nas amostras de exemplo, que seja capaz de rotular uma nova amostra sem rótulo que seja inserida posteriormente. Em síntese, esse tipo de aprendizado é utilizado quando as classes já são conhecidas pelo usuário.

- No **aprendizado não-supervisionado** as amostras de exemplos geralmente são vetores de características sem rótulos. Dessa maneira, sem conhecer a classe de cada amostra, o indutor procura agrupar as amostras de alguma maneira, formando agrupamentos ou *clusters*. Nesse caso, como as amostras não são rotuladas, é necessário que seja feita uma

análise para identificar o significado de cada grupo com base no contexto do problema. Em síntese, esse tipo de aprendizado é utilizado quando o usuário não tem conhecimento das classes possivelmente existentes.

A Figura 18 apresenta de forma geral a hierarquia do aprendizado indutivo onde no segundo nível temos a separação entre os modelos supervisionados e não-supervisionados e no terceiro nível temos alguns exemplos de modelos comumente utilizados.

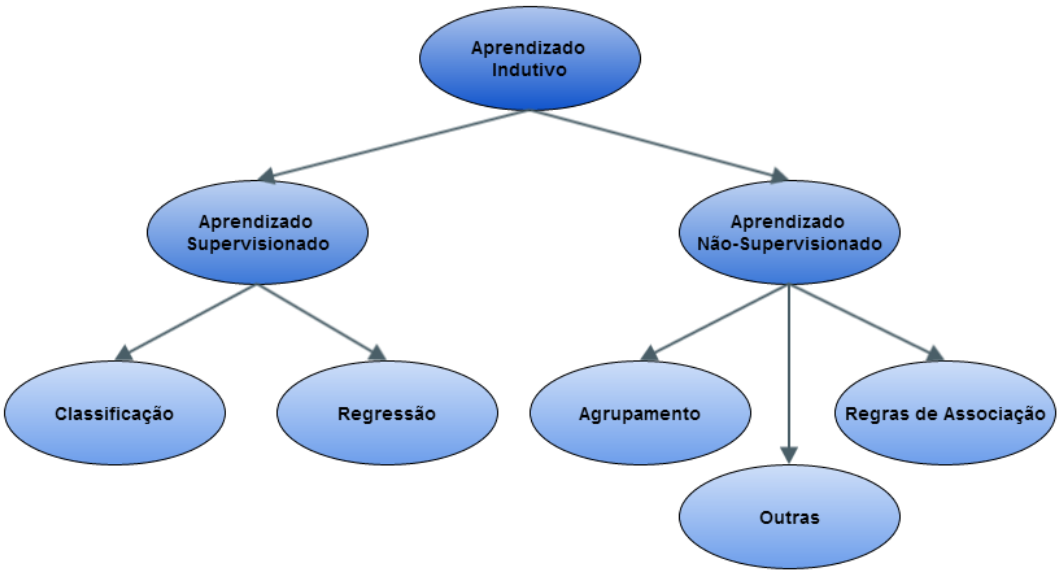


Figura 18: Visão geral da hierarquia do aprendizado indutivo
Fonte: Autoria própria.

Entre os métodos de aprendizado supervisionado existem os métodos de classificação. Os algoritmos de classificação utilizam uma função para gerar um modelo que possa classificar uma informação em uma das classes pré-definidas. Inicialmente o algoritmo utiliza um conjunto de dados de treinamento, onde todas as amostras possuem as características escolhidas e a indicação da classe a qual ela pertence. Durante a fase de treinamento o algoritmo passa a identificar um modelo mais adequado que seja capaz de relacionar o conjunto de atributos com os rótulos das classes, de modo que o modelo gerado seja capaz de classificar uma nova instância introduzida sem classe definida.

Existem diversos métodos diferentes de classificação que geram modelos diferentes, por exemplo:

- Classificadores Bayesianos que geram um modelo probabilístico baseado no Teorema de Bayes. Ex. *Naive Bayes*;

- Algoritmos de Árvore de Decisão que geram um modelo baseado em regras. Ex.: J48;
- Algoritmos de aprendizagem baseado em instâncias que geram um modelo baseado na proximidade das instâncias através de funções de similaridade. Ex.: K-vizinhos mais próximos (KNN);

Quanto mais informação sobre as classes as amostras possuírem, melhor será o resultado obtido pelo indutor. Não é a quantidade de atributos que fará o classificador ser mais eficiente, e sim o grau de informação que cada atributo carrega sobre a classe em questão (MONARD; BARANAUSKAS, 2003). O conhecimento do domínio pode ser utilizado por um especialista para realizar a escolha dos atributos ou fornecer alguma informação prévia como entrada.

A Figura 19 apresenta de forma geral um processo simples de classificação na qual um algoritmo de aprendizagem é utilizado para construção de um modelo com base nas instâncias do conjunto de treinamento cujos rótulos das classes são conhecidos. Após a construção do modelo, o mesmo é aplicado em um conjunto de teste onde os rótulos das classes são desconhecidos a fim de que o modelo seja capaz de predizer as classes para cada amostra.

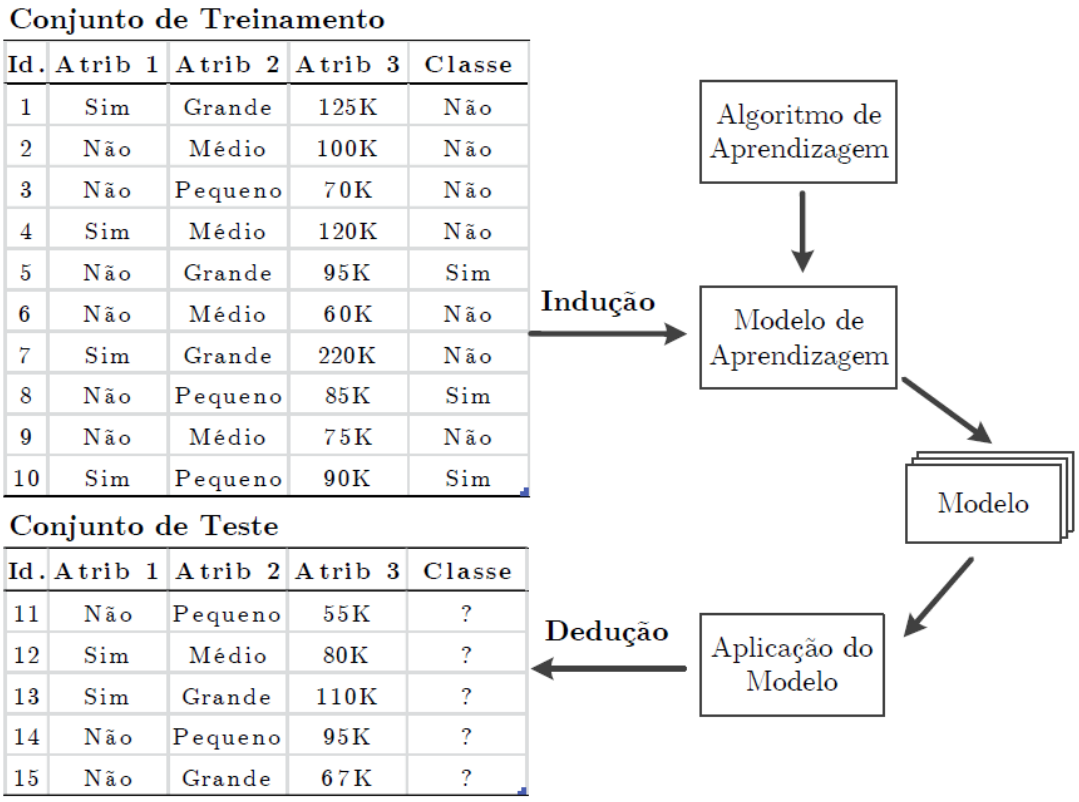


Figura 19: Visão geral da construção de um modelo de classificação

Fonte: (TAN et al., 2009)

3.3 PROBABILIDADE CONDICIONAL E TEOREMA DE BAYES

De acordo com Dougherty (2013, p. 43), se E, F, G, são eventos, a probabilidade de que esses eventos ocorram pode ser representada por um número real entre 0 e 1, representados por, $P(E)$, $P(F)$, $P(G)$.

A probabilidade de um evento está ligada à frequência relativa com que esse evento ocorre. Logo, se um experimento é observado por uma quantidade de vezes (N), e se o evento E ocorre M vezes, então temos $P(E) = M/N$.

Se E e F são eventos mutuamente exclusivos, então eles não podem ocorrer ao mesmo tempo. Dessa forma, a probabilidade da união de E e F ocorrer é representada por $P(E\ ou\ F)$ ou $P(E \cup F)$, que é dada pela equação:

$$P(E\ ou\ F) = P(E) + P(F) \tag{1}$$

Se os eventos E e F não são mutuamente exclusivos, ou seja, eles podem ocorrer simultaneamente, então temos que:

$$P(E\ ou\ F) = P(E) + P(F) - P(E\ e\ F) \tag{2}$$

Onde $P(E\ e\ F)$ ou $P(E \cap F)$ é a intersecção dos eventos. Geralmente chamada de **regra geral de adição**.

A soma de todas as probabilidades dos possíveis eventos é 1, logo se um evento E é certo que irá acontecer, a $P(E) = 1$, se for impossível de um evento ocorrer, a sua probabilidade será zero $P(E) = 0$. O complemento de um evento é tudo aquilo que não faz parte de E, e sua probabilidade $P(\overline{E})$ é dada por:

$$P(\overline{E}) = 1 - P(E)$$

A verificação dos eventos E e F serem independentes é dada por:

$$P(E \cap F) = P(E).P(F) \tag{3}$$

Considerando o exemplo de Dougherty (2013, p. 45), da independência dos eventos pelo lançamento de dois dados, onde a probabilidade do resultado do segundo dado lançado

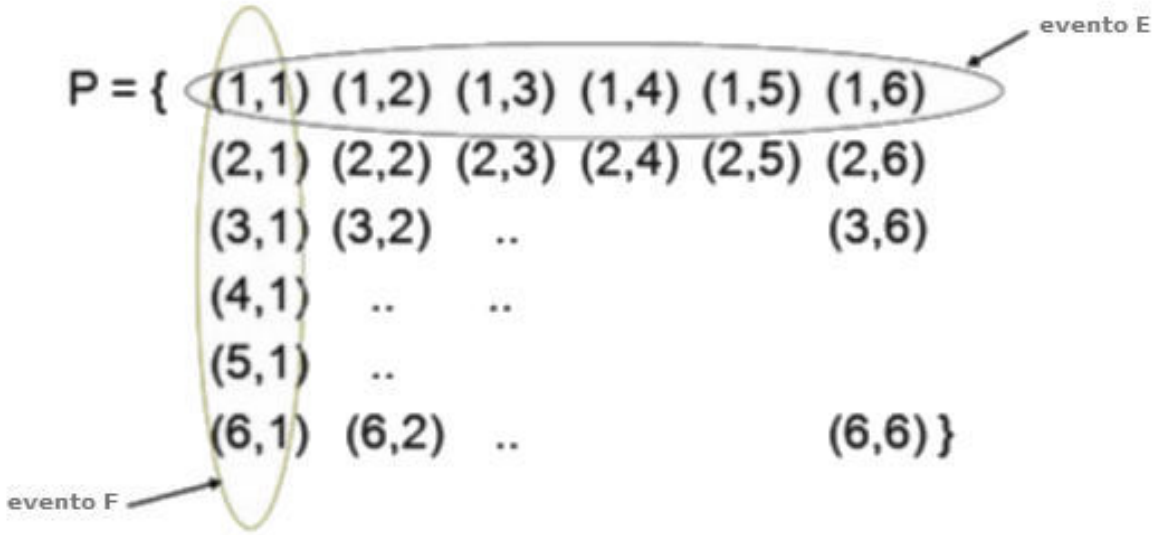


Figura 20: Espaço amostral para lançamento de dois dados, com evento E (resultado for "1") e evento F (resultado for "1") circulados em destaque

Fonte: Adaptado de (DOUGHERTY, 2013, p. 45)

não sofre nenhuma interferência pelo resultado do lançamento do primeiro dado. Temos que a probabilidade do evento E (lançamento do primeiro dado ter o resultado igual a "1") é $P(E) = 6/36 = 1/6$. E a probabilidade do evento F (lançamento do segundo ter o resultado igual a "1") é $P(F) = 6/36 = 1/6$. A Figura 20 apresenta o espaço amostral desse problema.

Logo, a probabilidade de qualquer dado ter o resultado igual a "1" é:

$$P(E\ ou\ F) = P(E) + P(F) - P(E\ e\ F) = 6/36 + 6/36 - 1/36 = 11/36$$

A **probabilidade condicional** é a probabilidade da ocorrência de um evento E, dado ocorrência de outro evento F. A probabilidade condicional entre os eventos E e F é dada por $P(E|F)$ e é lida como "a probabilidade de E, dado que F é verdadeiro".

Na prática só podemos falar em probabilidade condicional entre dois eventos se os dois eventos ocorrem simultaneamente. Logo, dado que o evento F tenha ocorrido, o que se obtém como probabilidade condicional destes eventos é o percentual que a probabilidade para a ocorrência de ambos os eventos E e F tem sobre a probabilidade de ocorrência do evento F. A probabilidade de F ser verdadeiro é $P(F)$, e a probabilidade de que E e F ocorreram é $P(E\ e\ F)$, então a probabilidade condicional de E dado que a ocorrência de F é verdadeira é obtida por:

$$P(E|F) = \frac{P(E\ e\ F)}{P(F)} \tag{4}$$

Assim temos também que a probabilidade condicional do evento F, dado que a ocorrência do evento E é verdadeira é dada por:

$$P(F|E) = \frac{P(E\ e\ F)}{P(E)} \tag{5}$$

Aplicando a multiplicação cruzada nas definições de probabilidade condicional nas Equações (4) e (5) obtêm-se a **regra do produto de probabilidades**.

$$P(E\ e\ F) = P(E|F).P(F) = P(F|E).P(E) \tag{6}$$

Manipulando a Equação 6, igualando os dois termos equivalentes à direita, e rearranjando obtêm-se o **Teorema de Bayes**:

$$P(E|F) = \frac{P(F|E).P(E)}{P(F)} \tag{7}$$

Onde a probabilidade de $P(E|F)$ é conhecida como **probabilidade posteriori** e pode ser expressa informalmente como:

$$posteriori = \frac{\text{verossimilhança}\ .\ priori}{\text{evidência}} \tag{8}$$

Se $\{E1, E2, E3, ..., En\}$ é um conjunto de eventos mutualmente exclusivos que juntos formam o espaço S, então $P(F)$ é constante para cada um deles. Nesse caso:

$$P(E|F) = \frac{P(F|E).P(E)}{\sum P(F|E_i).P(E_i)} \tag{9}$$

Para o caso de uma partição binária onde o espaço S é composto por $\{E, \overline{E}\}$, então:

$$P(E|F) = \frac{P(F|E).P(E)}{P(F|E).P(E) + P(F|\overline{E}).P(\overline{E})} \tag{10}$$

O **Teorema de Bayes** permite descobrir a probabilidade a *posteriori* $P(E|F)$ de um determinado evento E ocorrer dado a ocorrência do evento F a partir do conhecimento de estimativas das probabilidades prévias $P(E)$ e $P(F)$ e do conhecimento da probabilidade condicional

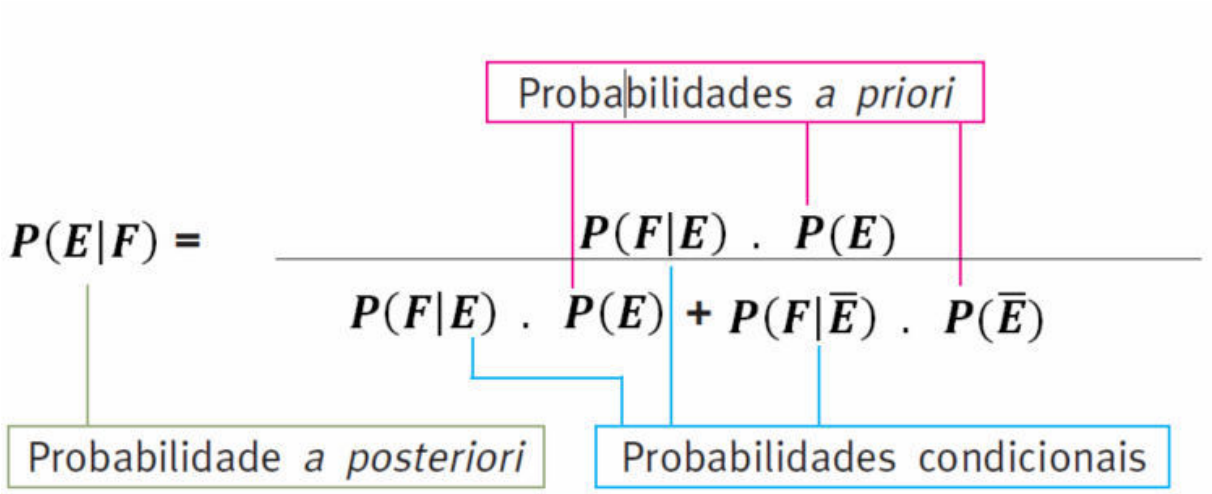


Figura 21: Fórmula da equação do Teorema de Bayes para um espaço amostral binário com a origem de cada informação destacada

Fonte: Adaptado de (PENA, 2006)

$P(F|E)$, como pode ser observado na Figura 21.

A aplicação prática do Teorema de Bayes pode ser vista no exemplo dado por Dougherty (2013, p. 51), que trata sobre um caso de diagnóstico de câncer de mama. As medidas tradicionais utilizadas como valor de teste em diagnóstico são a **Sensitividade** (probabilidade condicional do teste identificar aqueles com a doença dado que eles tenham a doença) e **Especificidade** (probabilidade condicional do teste identificar aqueles livres da doença dado que eles não tenham a doença).

$$sensitividade,\ P(F|E) = VP/(VP+FN)$$

$$especificidade,\ P(\overline{F}|\overline{E}) = VN/(VN/FP)$$

Nesse exemplo são assumidos que:

- Cerca de 1% das mulheres que participam do exame de rotina de câncer de mama possuem câncer;
- Cerca de 80% daquelas com câncer recebem um resultado positivo no teste e 9.6% sem câncer também recebem um resultado positivo.

Supondo que uma mulher receba um resultado positivo. Qual a probabilidade dela realmente ter câncer de mama?

Normalmente as pessoas costumam errar a resposta dessa e de outras questões semelhantes, pois não consideram a probabilidade a priori do problema, que nesse caso é 1% (muito

baixa). Ao se deparar com a informação de que 80% das pessoas diagnosticadas positivamente possuem câncer, intuitivamente pensam que a probabilidade da pessoa de fato ter câncer é alta. Uma forma de resolver esse problema é através da utilização do teorema de Bayes para calcular a probabilidade a posteriori. A seguir é apresentada a forma de resolução do problema através da utilização do Teorema de Bayes.

A mulher que obtém um resultado positivo na mamografia e realmente tem câncer de mama equivale a 80% de 1%, ou 0,8%, de todas as mulheres que foram testadas. A mulher que não tem câncer de mama, mas continua a apresentar um resultado (falso) positivo na mamografia, equivale a 9,6% de 99%, ou 9,504%, de todas as mulheres que foram testadas. Por tanto, o total de resultados positivos na mamografia equivale a 0,8% + 9,504%, ou 10,304%, de todas as mulheres que foram testadas. Dessa porcentagem, a mulher que realmente possui câncer (0,8%) é a distinta minoria.

A probabilidade de uma mulher que obtém um resultado positivo no teste realmente possuir o câncer de mama, $P(E|F)$, é $0,8/10.304 = 0,0776$ ou $7,76\%$.

De forma alternativa podemos formalizar o problema como:

$E =$ possuir câncer de mama

$F =$ teste positivo

$P(E) = 0,01$

Sensitividade $P(F|E) = 0,8$ $P(F|\overline{E}) = 0,096$

Por tanto,

Especificidade $P(\overline{F}|\overline{E}) = 1 - 0,096 = 0,904$ *e* $P(\overline{E}) = 1 - 0,01 = 0,99$

Utilizando $P(E\ e\ F) = P(E|F).P(F)$;

$P(F\ e\ E) = P(F|E).P(E) = 0.8.0.01 = 0.008$

$P(F\ e\ \overline{E}) = P(F|\overline{E}).P(\overline{E}) = 0,096.0,99 = 0,09504$

A probabilidade a posteriori de realmente possuir câncer de mama é:

$$\begin{aligned} P(E|F) &= P(E\ e\ F)/P(F) \\ &= P(E\ e\ F)/(P(F\ e\ E) + P(F\ e\ \overline{E})) \\ &= 0,008/0,10304 \\ &= 0,07764 \\ &= 7,76\% \end{aligned}$$

Com base na Teoria de Bayes que usa a probabilidade a priori e condicional, surgiram os classificadores Bayesianos que são apresentados na Seção 3.5 desse capítulo.

3.4 REDES BAYESIANAS

As redes Bayesianas são atualmente uma das abordagens mais promissoras para o processo de descoberta de conhecimento em base de dados (SEBASTIANI et al., 2010).

De acordo com Sebastiani et al. (2010), as redes Bayesianas pertencem a uma classe mais geral de modelos chamados de modelos probabilísticos gráficos que surgiram da combinação da teoria dos grafos e da teoria da probabilidade. O seu sucesso está na habilidade de lidar com modelos probabilísticos complexos através da decomposição em componentes menores e passíveis.

Segundo Sebastiani et al. (2010), um modelo probabilístico gráfico é definido por um grafo onde os nós representam variáveis estocásticas e os arcos representam as dependências entre tais variáveis. Estes arcos são construídos por distribuição de probabilidade moldando a interação entre as variáveis ligadas.

Um modelo gráfico probabilístico é chamado de rede Bayesiana quando o grafo conectando suas variáveis é um Grafo Acíclico Dirigido (DAG - *Directed Acyclic Graph*). Este grafo representa suposições de independência condicional que são usados para fatorar a distribuição de probabilidade conjunta das variáveis de rede, tornando assim o processo de aprendizagem de grandes bancos de dados passíveis de serem calculados. Uma rede Bayesiana induzida de informações pode ser utilizada para investigar relacionamentos distantes entre variáveis, assim como fazer predições e explicações pelo cálculo da probabilidade de distribuição condicional de uma variável, dados os valores de outras (SEBASTIANI et al., 2010).

De acordo com Friedman et al. (1997), trabalhos clássicos em aprendizado supervisionado mostram que surpreendentemente um classificador Bayesiano simples, assumindo a independência entre as características, é competitivo com os classificadores atuais, tal como o C4.5.

3.5 CLASSIFICADORES BAYESIANOS

Segundo Friedman et al. (1997), classificação é uma tarefa básica na análise de dados e reconhecimento de padrões que requer a construção de um classificador, ou seja, uma função que encontre a qual classe uma instância pertence baseada no conjunto de seus atributos.

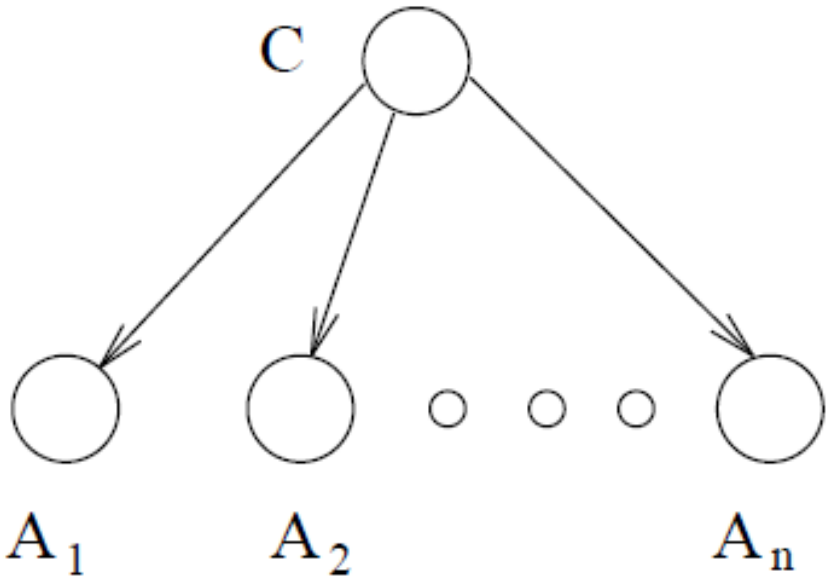


Figura 22: Estrutura de um classificador *Naive Bayes*
Fonte: (FRIEDMAN et al., 1997)

De acordo com Dougherty (2013, p. 53), o classificador Naive Bayes (NB) é um classificador probabilístico simples que é baseado na aplicação do Teorema de Bayes apresentado na Equação 7, assumindo que todas as características são independentes entre si.

Segundo Flores et al. (2011), a abordagem do NB é confiável, pois não é necessário um grande conjunto de treinamento para se obter uma estimação de probabilidade aceitável.

Em função da restrição na topologia da rede, apresentada na Figura 22, a etapa de treinamento do classificador NB consiste na estimação da distribuição de probabilidade condicional para cada atributo, dado a classe, a partir de um conjunto de treinamento. Uma vez treinado, o NB classifica as instâncias através do cálculo da distribuição de probabilidades a posteriori sobre a classe através do Teorema de Bayes e associa as instâncias à classe com a maior probabilidade a posteriori (SEBASTIANI et al., 2010).

Quando representado por uma rede Bayesiana, o NB possui a estrutura descrita na Figura 22. Essa rede representa o principal pressuposto por trás do classificador NB, de que cada atributo é independente entre si, dado a sua classe.

Segundo Jiang et al. (2007), este classificador utiliza um conjunto de treinamento para aprender a probabilidade condicional de cada atributo A_i dado o rótulo da classe C. A classificação é então feita aplicando a regra de Bayes para calcular a probabilidade de C dada a instância de A_1, ..., A_n, e então prediz a classe com a maior probabilidade a posteriori. Este cálculo é baseado em uma suposição de independência onde os atributos A_i são condicional-

mente independentes, dado o valor da classe C.

Dadas as seguintes definições:

- A_1, A_2, ..., A_n são n atributos (correspondendo ao nó de atributo em uma rede Bayesiana);
- t é uma instância representada pelo vetor $(a_1, a_2, a_3, ..., a_n)$ onde a_i é o valor de A_i;
- C é o conjunto de rótulos ou classes (correspondendo ao nó de classe em uma rede Bayesiana);
- c representa o valor que C assume e $c(t)$ para denotar a classe de t.

O classificador Bayesiano representado por uma rede Bayesiana é definido na Equação 11.

$$c(t) = arg\ max\ P(c)P(a_1, a_2, a_3, ..., a_n|C) \tag{11}$$

Assumindo que todos os atributos são independentes, dada a classe, chamado de suposição de independência condicional, temos a Equação 12.

$$P(t|c) = P(a_1, a_2, a_3, ..., a_n|c) = \prod_{i=1}^{n} P(a_i|c) \tag{12}$$

Então o classificador resultante é chamado classificador Bayesiano ingênuo, ou simplesmente NB, Equação 13.

$$c(t) = \arg \max_{c \in C} P(c) \prod_{i=1}^{n} P(a_i|c) \tag{13}$$

O desempenho do NB é um tanto surpreendente dado que a suposição de independência entre os atributos é quase sempre irreal (FRIEDMAN et al., 1997).

Diante da limitação de independência dos atributos da rede Bayesiana ingênua, diversos trabalhos foram realizados a fim de aperfeiçoar os resultados obtidos por esse modelo.

Jiang et al. (2007), afirma que embora o *Naive Bayes* seja fácil de ser construído devido ao valor de $P(A|C)$ poder ser facilmente estimado a partir das amostras de treinamento, a suposição de independência condicional feita pela abordagem ingênua, prejudica o desempenho da classificação do NB quando isso é violado.

A fim de relaxar essa suposição de modo eficiente, é necessária uma linguagem apropriada e máquinas eficientes para representar e manipular as afirmações de independência. Chickering (1996) provou que a aprendizagem de uma rede Bayesiana é um problema NP-completo e a fim de evitar a complexidade inacessível para a aprendizagem das redes Bayesianas diversos pesquisadores têm desenvolvido formas de aperfeiçoamento para os classificadores Bayesianos.

Esses trabalhos de aperfeiçoamento para os classificadores Bayesianos podem ser divididos de maneira ampla em quatro abordagens de acordo com Jiang et al. (2005):

- **Seleção de características:** Selecionando conjuntos de atributos no qual os atributos satisfaçam a suposição de independência dos mesmos;
- **Extensão de estrutura:** Estendendo a estrutura do NB para representar as dependências através dos atributos;
- **Aprendizagem local:** Empregando o princípio do aprendizado local para encontrar um conjunto de treinamento local e utilizar este para construir o NB;
- **Expansão de dados:** Expandindo os dados de treinamento e construir o NB utilizando os dados de treinamento expandidos.

Dada a vasta literatura sobre as extensões dos classificadores Bayesianos, neste trabalho serão analisados apenas os métodos que realizam a extensão de estrutura, visto que no problema sendo abordado neste trabalho, existe uma possível relação entre os atributos que serão utilizados nos experimentos.

Conforme descrito anteriormente, o NB possui uma suposição de que os atributos são condicionalmente independentes, porém nas aplicações de aprendizado de máquina reais essa suposição é irreal. Assim, estendendo a estrutura do NB e utilizando arcos para explicitamente representar as dependências entre os atributos é uma forma direta de relaxar essa suposição. Como resultado é gerado um classificador Bayesiano. Contudo, como dito anteriormente aprender a estrutura ótima para uma rede é um problema NP-completo. Na prática, é necessário que sejam impostas restrições à estrutura da rede Bayesiana.

Como exemplo, Friedman et al. (1997) apresentam o algoritmo de aprendizado *Tree Augmented Naive Bayes* (TAN) que consegue aprender um conjunto ótimo em uma complexidade computacional em tempo polinomial e em alguns experimentos obteve resultados melhores comparado ao Naive Bayes e ao C4.5.

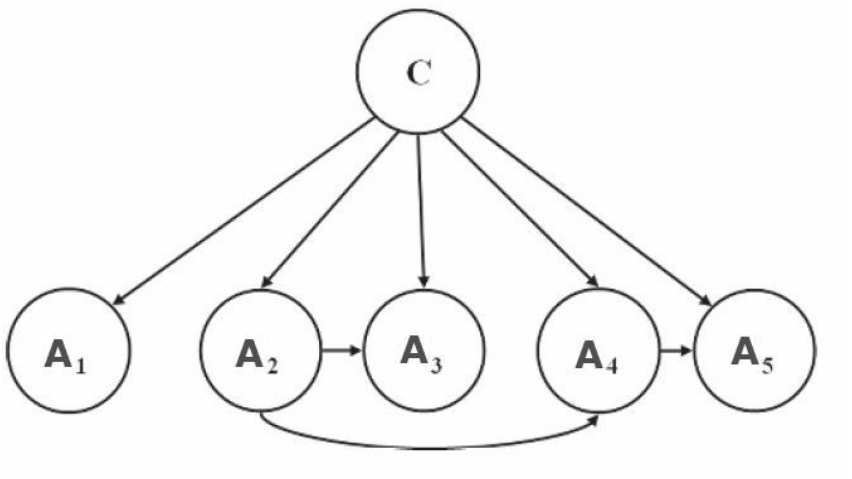


Figura 23: Estrutura de um classificador TAN
Fonte: Adaptado de (SEBASTIANI et al., 2010)

De acordo com Sebastiani et al. (2010), o TAN talvez seja um dos classificadores mais competitivos para relaxar a suposição de que os atributos sejam independentes. No TAN, cada atributo tem uma variável classe como pai e também podem ter no máximo um outro atributo como pai. Para evitar ciclos, os atributos precisam estar ordenados e o primeiro atributo não pode ter outros pais a não ser uma variável de classe. A Figura 23 apresenta a estrutura de um classificador TAN contendo cinco atributos.

Um algoritmo para inferir um classificador TAN precisa escolher tanto a estrutura de dependência entre os atributos quanto o parâmetro que quantifica essa dependência. Em função da simplicidade em sua estrutura, a identificação de um classificador TAN não necessita de busca, mas da construção de uma árvore entre os atributos. Um algoritmo chamado *Construct-TAN* (CTAN) foi proposto em (FRIEDMAN et al., 1997) e uma das limitações do CTAN é que é aplicável apenas para atributos discretos, de modo que qualquer atributo contínuo necessita ser primeiramente discretizado (SEBASTIANI et al., 2010).

Outra extensão dos classificadores Bayesianos foi apresentada por Sahami (1996), a fim de relaxar as suposições de independência condicional entre os atributos do NB e outras suposições existentes no TAN. Sahami apresentou um algoritmo chamado *K-Dependence Bayesian Classifier* (KDBC), onde o valor K representa o número máximo de nós pais que um atributo pode ter além da classe. Dessa forma um 0-DBC é equivalente ao Naive Bayes e o 1-DBC é equivalente a um classificador TAN. A vantagem desse classificador em relação ao TAN é a sua flexibilidade. No TAN, a variável pode ter no máximo uma característica associada a ela além da variável de classe. Essa restrição do número de pais limita as dependências que podem ser modeladas entre grupos de características (FLORES et al., 2011).

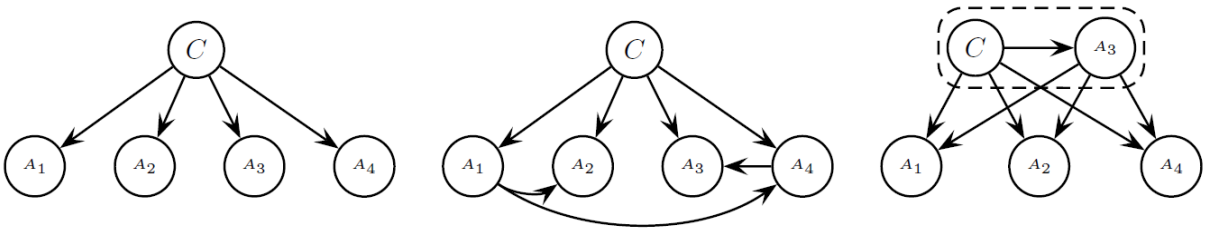
Outros trabalhos também foram realizados na tentativa de diminuir a suposição de independência dos atributos, e alguns deles demonstraram acurácia notável. *Lazy Bayesian Rules* (LBR), (ZHENG; WEBB, 2000) e *Super-Parent TAN* (SP-TAN), (KEOGH; PAZZANI, 1999), uma variação do *Tree Augmented Naive Bayes* (TAN), (FRIEDMAN et al., 1997). Porém essas duas técnicas possuem alto custo computacional, SP-TAN tendo alta complexidade computacional durante a etapa de treinamento e LBR tendo alta complexidade computacional durante a etapa de classificação. Ao analisar o LBR e o SP-TAN os autores identificaram os dois principais fatores de geração de carga computacional como sendo a seleção de modelo e geração de estimativa de probabilidade em tempo real para o LBR e estimativa de probabilidade via tabela de probabilidade de 3 dimensões para o SP-TAN. Esses custos reduzem a utilidade desses métodos como uma alternativa ao NB.

Como alternativa Webb et al. (2005) apresenta um modelo chamado Averaged One-Dependence Estimators (AODE), uma técnica eficiente que utiliza uma suposição de independência de atributo mais fraca do que o NB, melhorando a acurácia de previsão sem gerar custo computacional indevido.

Segundo Flores et al. (2011), para manter a eficiência, o AODE é restrito ao uso exclusivo de um único estimador de dependência. Especificamente, o AODE pode ser considerado um conjunto de *Superparent One-Dependence Estimators* (SPODEs), porque cada atributo depende da classe e de outro atributo compartilhado, designado como superpai.

Graficamente, todos os classificadores utilizados no AODE terão uma estrutura como a apresentada na Figura 24 (c), onde AODE combina todas as possibilidades de classificadores com esse padrão de estrutura.

Na Figura 24 são apresentadas as estruturas das redes Bayesianas geradas pelos classificadores Naive Bayes (0-DBC), TAN (1-DBC) e AODE, respectivamente, onde é possível visualizar as diferenças estruturais de topologia para cada um dos casos.



(a) Naive Bayes (b) TAN/KDB1 (c) AODE

Figura 24: Exemplos de estruturas de rede com quatro atributos preditivos para os seguintes classificadores de rede Bayesiana: NB, TAN e KDB1, AODE

Fonte: Adaptado de (FLORES et al., 2011)

3.6 MÉTODOS PARA A AVALIAÇÃO DE MODELOS

No processo do KDD, mais especificamente na etapa de mineração de dados, o resultado obtido pela utilização de um algoritmo no conjunto de dados é um modelo. Comumente é utilizado mais de um algoritmo de mineração no mesmo conjunto de dado, cada um produzindo seu respectivo modelo, a fim de obter o melhor resultado com o menor custo computacional.

Não existe um único algoritmo que apresente um melhor desempenho para todos os problemas, (MONARD; BARANAUSKAS, 2003). Com isso se faz necessária a utilização de técnicas de avaliação para aferir o grau de eficácia de cada modelo.

Nesta seção são apresentadas algumas técnicas utilizadas para avaliação dos classificadores e que serão utilizadas para avaliação dos experimentos realizados neste trabalho.

3.6.1 MATRIZ DE CONFUSÃO

Segundo Kohavi e Provost (1998), a matriz de confusão apresenta os resultados esperados e os obtidos de uma classificação. A mesma possui tamanho *LxL*, onde *L* é o número de valores de classes diferentes.

A matriz permite uma visualização inequívoca dos resultados da classificação e por isso é amplamente utilizada para avaliação de resultados. É composta por duas entradas, uma das entradas é a classe esperada pelo modelo e a outra é a classe prevista. As células são preenchidas com o número de instâncias referente ao cruzamento das duas entradas.

Na Figura 25 é apresentado um exemplo de matriz de confusão onde as entradas verticais representam as classes obtidas pelo modelo e as entradas horizontais as classes originais

do conjunto de treinamento. Podemos ver que no caso da classe A, todas as instâncias foram corretamente classificadas. Já no caso da classe B o modelo acertou a previsão de 90 instâncias, mas classificou incorretamente 10 instâncias como sendo pertencentes à classe C. No caso da classe C todas as instâncias foram corretamente classificadas.

A	**B**	**C**	
100	0	0	**A**
0	90	10	**B**
0	0	100	**C**

Figura 25: Ilustração de uma tabela de exemplo de matriz de confusão

Fonte: Autoria própria.

A partir da matriz de confusão podem-se obter outras métricas para aferir do desempenho do modelo de classificação.

Tomando como exemplo uma matriz de confusão binária apresentada na Figura 26 onde as amostras pertencentes à classe são rotuladas como "+" (positivas), enquanto as amostras não pertencentes à classe são rotuladas como "-" (negativas).

	Predito C_+	**Predito** C_-
C_+	V_p	F_n
C_-	F_p	V_n

Figura 26: Matriz de confusão para duas classes

Fonte: Autoria própria.

Onde temos:

V_p = Número de amostras positivas da classe que foram previstas corretamente.

F_n = Número de amostras negativas da classe que foram previstas incorretamente.

F_p = Número de amostras positivas da classe que foram classificadas incorretamente.

V_n = Número de amostras negativas da classe que foram previstas corretamente.

Através desses indicadores contidos na matriz de confusão podem-se obter diversas medidas de desempenho, entre elas:

- **Acurácia:** porcentagem de amostras positivas e negativas classificadas corretamente sobre a soma de amostras positivas e negativas.

$$\text{Acurácia} = \frac{V_p + V_n}{V_p + V_n + F_p + F_n}$$

- **Precisão:** porcentagem de amostras positivas classificadas corretamente sobre o total de amostras classificadas como positivas.

$$\text{Precisão} = \frac{V_p}{V_p + F_p}$$

- **Sensitividade (Recall):** porcentagem de amostras positivas classificadas corretamente sobre o total de amostras positivas.

$$\text{Recall} = \frac{V_p}{V_p + F_n}$$

- **F-measure** também chamada de **F-score:** é a média ponderada de precisão e *recall*.

$$\text{F-measure} = 2.\frac{\text{Precisão . Recall}}{\text{Precisão} + \text{Recall}}$$

No contexto de conjunto de dados balanceado e custo de erro equalizado, é sensato utilizar a precisão e a taxa de erro como métricas de desempenho. Taxa de erro é 1 - Acurácia.

Na presença de conjunto de dados desbalanceados e custo de erro desigual, é mais apropriado utilizar a curva ROC ou outra técnica semelhante (CHAWLA et al., 2002).

Quando o conjunto de treinamento possui amostras desbalanceadas, é desejável utilizar uma medida de desempenho diferente da precisão, pois nesse caso a medida de precisão pode não refletir o real resultado do modelo. Por exemplo, considere um conjunto de treinamento com duas classes (C_1, C_2) e a seguinte distribuição (99%, 1%).

Essa situação pode ser indesejável quando a classe minoritária representa informação importante, por exemplo:

C_1 = Baixo risco;

C_2 = Alto risco.

Nesse caso um classificador simples que classifique as novas instâncias sempre como sendo da classe majoritária C_1 teria uma precisão de 99,00%, mas não seria capaz de classificar as amostras mais relevantes para o problema que se encontram na classe minoritária.

Na curva ROC o eixo X representa *FP = FP / (VN + FP)* e eixo Y representa *VP = VP / (VP + FN)*. O ponto ideal na curva ROC seria (0,100), que significa que todos os exemplos positivos foram classificados corretamente e nenhum exemplo negativo foi classificado erroneamente como positivo. A área abaixo da curva ROC (AUC) é uma medida útil para desempenho

de classificador de modo que essa medida é independente do critério de seleção e probabilidade a priori (CHAWLA et al., 2002).

3.6.2 *CROSS-VALIDATION*

Segundo MONARD e BARANAUSKAS (2003), este é um tipo de estimador por amostragem. Em *r-fold cross-validation*, as amostras são aleatoriamente divididas em *r* partições mutuamente exclusivas chamadas de *folds*. Essas partições possuem tamanho aproximadamente igual a (n/r) amostras. As amostras nas partições $(r-1)$ são usados para treinamento e a hipótese induzida é testada na partição remanescente. Este processo é repetido *r* vezes, cada vez considerando uma partição diferente para teste. O erro deste estimador é a média dos erros calculados em cada um dos *r folds*.

3.6.3 *LEAVE-ONE-OUT*

MONARD e BARANAUSKAS (2003) descrevem o estimador *leave-one-out* como um caso especial de *cross-validation*. Os autores afirmam que esse método é computacionalmente dispendioso e frequentemente é usado em amostras pequenas. Para uma amostra de tamanho *n* uma hipótese é induzida utilizando $(n-1)$ amostras. A hipótese é então testada na única amostra deixada de fora do conjunto. Esse processo é repetido *n* vezes cada vez deixando de considerar uma única amostra para ser testada após a indução. O erro é a soma dos erros em cada teste dividido por *n*.

3.7 DCG (*DISCOUNTED CUMULATIVE GAIN*)

Além de identificar corretamente as famílias em situação de vulnerabilidade e risco social é importante que exista uma ordenação das famílias que necessitam ser prioritariamente atendidas para que os técnicos da assistência social possam atender essas famílias de forma ordenada. Nesse trabalho foram utilizados métodos de classificação probabilísticos que permitem identificar o grau de probabilidade de uma amostra pertencer a uma determinada classe, neste caso, o grau de probabilidade de uma família estar em alta vulnerabilidade e risco social. Através desse indicador de probabilidade pode-se gerar uma lista ordenada em formato de *ranking* para que as famílias mais vulneráveis sejam atendidas primeiramente. Para avaliar qual configuração experimental é capaz de gerar a melhor lista ordenada das famílias foi utilizada uma medida de qualidade de *ranking* chamada DCG.

Discounted Cumulative Gain (DCG), proposta por JÄrvelin e KekÄlÄinen (2002), é

uma medida de qualidade de *ranking*. É comumente utilizada para medir a eficiência de algoritmos de busca na web e outras aplicações relacionadas. Utilizando uma escala de relevância dos documentos retornando em uma lista por um mecanismo de busca, o DCG mede a utilidade de um documento baseado em sua posição na lista. O ganho é acumulado desde o topo até o final da lista de resultados com o ganho de cada resultado descontado conforme a sua posição baixa.

Quando se examina uma lista de resultados ordenados por uma consulta, duas suposições são consideradas:

1. Documentos altamente relevantes são mais úteis do que documentos marginalmente relevantes.

2. Quanto maior for a posição de um documento relevante no *ranking*, menor será o valor dele para seu usuário. Os documentos altamente relevantes são mais úteis quando aparecem primeiro em uma lista ordenada.

Segundo JÄrvelin e KekÄlÄinen (2002), o **DCG** origina-se de uma medida anterior e mais primitiva chamada **CG** (*Cumulative Gain*) que não inclui a posição de um resultado em consideração na avaliação de utilidade do resultado.

O valor CG de uma posição particular de p é definida como:

$$CG_p = \sum_{i=1}^{p} rel_i \tag{14}$$

onde rel_i é a relevância do resultado na posição i.

A premissa do **DCG** é que itens altamente relevantes que aparecerem mais baixo no resultado da busca devem ser penalizados da forma que o valor da relevância é reduzido de forma logarítmica proporcional a posição do resultado.

A medida DCG de uma posição particular p é definida como:

$$DCG_p = rel_1 + \sum_{i=2}^{p} \frac{rel_i}{log_2(i)} \tag{15}$$

Como exemplo, baseado em JÄrvelin e KekÄlÄinen (2002), temos uma lista de seis documentos retornados por um mecanismo de busca. Para cada documento um avaliador indicou uma pontuação de relevância variando entre 0, 1, 2 e 3, sendo 3 para documentos altamente relevantes e 0 para documentos irrelevantes.

$D_1, D_2, D_3, D_4, D_5, D_6$

Após configurar a relevância de cada documento o resultado é:

$3, 2, 3, 0, 1, 2$

O ganho acumulado (CG) dessa lista de resultados é:

$$CG_6 = \sum_{i=1}^{6} rel_i = 3+2+3+0+1+2 = 11 \tag{16}$$

Ao trocar a ordem de qualquer dois documentos o valor do CG não é afetado. Se os documento D_3 e D_4 forem trocados, o CG continuaria ter o mesmo valor, 11. O DCG é utilizado para enfatizar o aparecimento de documentos mais relevantes no topo da lista. Utilizando uma escala de redução logarítmica, o DCG para cada resultado em ordem é apresentado na Tabela 2:

Tabela 2: Valores DCG

i	rel_i	$\log_2 i$	$\frac{rel_i}{\log_2 i}$
1	3	0	N/A
2	2	1	2
3	3	1,585	1,892
4	0	2,0	0
5	1	2,322	0,431
6	2	2,584	0,774

Fonte: Autoria própria.

Dessa maneira, o DCG_6 para essa lista ordenada é:

$$DCG_6 = rel_1 + \sum_{i=2}^{6} \frac{rel_i}{log_2(i)} = 3 + (2 + 1,892 + 0 + 0,431 + 0,774) = 8,10 \tag{17}$$

Se fosse realizada uma troca de ordem dos documentos D_3 e D_4, isto resultaria em um valor menor do DCG, porque um documento menos relevante seria posicionada primeiramente no *ranking*, isto é, um documento mais relevante é descontado mais por estar posicionada em um ponto mais baixo do *ranking*.

3.8 WEKA

O WEKA[1] (*Waikato Environment for Knowledge Analysis*) é uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados. Essa ferramenta começou a ser escrita em 1993, na Universidade de Waikato. Seus algoritmos podem ser aplicados diretamente a um conjunto de dados ou podem ser chamados através de outras aplicações desenvolvidas em código JAVA (HALL et al., 2009).

De acordo com Salama et al. (2012), o WEKA contém ferramentas para pré-processamento, classificação, regressão, agrupamento, regras associação, visualização e seleção de características. É uma ferramenta de código aberto (*Open Source*) sob licença pública GPL (*General Public License*) e pode ser utilizada livremente para realização de estudos.

Neste trabalho a biblioteca do WEKA foi utilizada como base para implementação de um aplicativo JAVA para realizar as etapas de pré-processamento, classificação e avaliação dos resultados.

[1] http://www.cs.waikato.ac.nz/ml/weka/

4 MINERAÇÃO DE DADOS DESBALANCEADOS

Segundo Hu et al. (2009) a classificação é uma importante questão na mineração de dados, aprendizagem de máquina e reconhecimento de padrões. Diversos algoritmos foram desenvolvidos e aperfeiçoados atingindo sucesso em aplicações em problemas reais. Porém quando esses algoritmos são utilizados em conjuntos de dados desbalanceados o desempenho para a classe minoritária tende a não ser satisfatório.

De acordo com Batista et al. (2004), a maioria dos sistemas de aprendizagem assumem que o conjunto de dados utilizado na etapa de treinamento é balanceado. Entretanto este não é sempre o caso em dados reais onde uma das classes pode ser representada por um grande número de amostras, enquanto a outra é representada por poucas amostras. Isto é conhecido como problema de classes desbalanceadas e é comumente reportado como um obstáculo na indução de bons classificadores através de algoritmos de aprendizado de máquina.

Segundo Han et al. (2005), em mineração de dados um conjunto de dados é considerado desbalanceado quando uma das classes possui uma quantidade de amostras muito inferior em relação às demais classes. A classe com a menor quantidade de amostras é chamada de minoritária e a classe com maior quantidade majoritária. A classe minoritária é geralmente a de maior interesse do ponto de vista de tarefas de aprendizagem. Esse é um problema comum em mineração de dados e diversos trabalhos do mundo real se deparam com essa situação, como por exemplo, detecção de fraudes em chamadas telefônicas (FAWCETT; PROVOST, 1997), (HILAS; MASTOROCOSTAS, 2008), diagnósticos de doenças (COHEN et al., 2006), classificação de texto (ZHENG et al., 2004) e detecção de poços de óleo através de imagens de satélite (KUBAT et al., 1998). O problema do desbalanceamento tem sido fortemente estudado e é um desafio atual para a mineração de dados.

Batista et al. (2004) explica de forma simples porque a aprendizagem a partir de conjuntos de dados desbalanceados tende a ser problemático. Imaginando a situação ilustrada na Figura 27. Na Figura 27 (a) existe uma grande desbalanceamento entre a classe majoritária (-) e a classe minoritária (+), e os dados apresentam certo grau de sobreposição. Uma situação

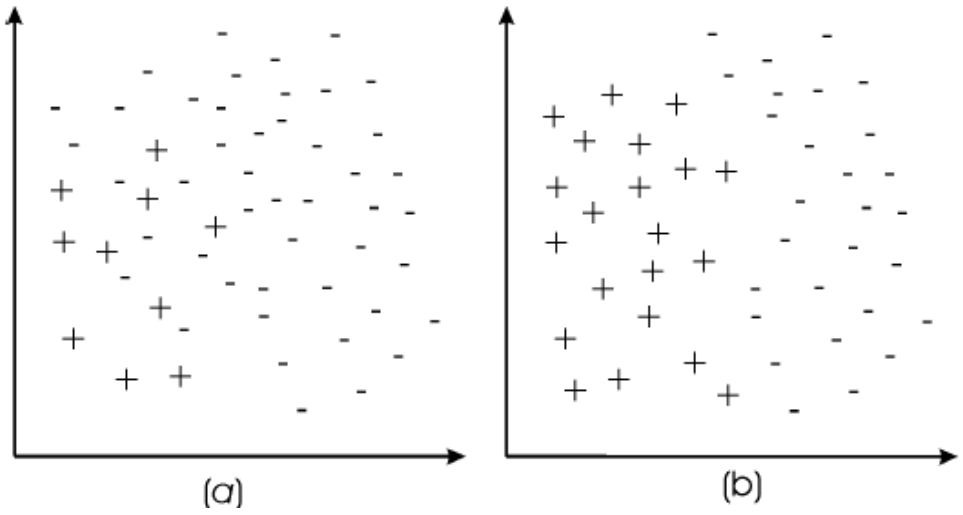


Figura 27: Várias amostras da classe majoritária (negativas) com algumas amostras da classe minoritária (positivas) esparramadas (a) conjunto de dados balanceado com agrupamentos bem definidos (b)

Fonte: (BATISTA et al., 2004)

muito mais confortável para a aprendizagem é apresentada na Figura 27 (b), onde as classes são balanceadas com agrupamentos bem definidos.

Em uma situação similar a ilustrada na Figura 27 (a), casos isolados da classe minoritária podem confundir um classificador como *k* vizinhos mais próximos (K-NN). Por exemplo, 1-NN deve classificar errado muitos casos da classe minoritária (+) porque os vizinhos mais próximos são pertencentes à classe majoritária (-). Em uma situação onde o desbalanceamento é muito alto, a probabilidade do vizinho mais próximo de um caso da classe minoritária ser um caso da classe majoritária tende a ser alto, e a taxa de erro da classe minoritária tenderá a ter valores altos, o que é inaceitável.

As árvores de decisão também sofrem um problema similar. Na presença de sobreposição de classes, as árvores de decisão podem precisar criar muitos testes para distinguir os casos da classe minoritária dos casos da classe majoritária. Podar a árvore de decisão pode não resolver o problema. Isto ocorre pois a poda remove alguns ramos considerados muito especializados, nomeando novos nós folha com a classe dominante nesse nó. Assim, existe uma alta probabilidade de que a classe majoritária seja também a classe dominante dos nós folhas.

Diversas abordagens foram criadas para tratar o problema do desbalanceamento e basicamente são divididas por Ramentol et al. (2012) em duas formas de soluções: **à nível de dados** (*data level*) que consiste no balanceamento da distribuição das classes através da criação

ou remoção de amostras, como feito em (CHAWLA et al., 2002) e (STEFANOWSKI; WILK, 2008); e **à nível de algoritmos** (*algorithmic level*) neste caso é realizado uma adaptação no algoritmo de aprendizagem para lidar diretamente com o desbalanceamento das amostras entre as classes, como feito em (GRZYMALA-BUSSE et al., 2005) e (WEISS; PROVOST, 2003).

Em diversos trabalhos tem sido mostrado que a aplicação de pré-processamento a fim de balancear a distribuição de classes é uma solução positiva para o problema de conjuntos de dados desbalanceados (BATISTA et al., 2004) e (FERNÁNDEZ et al., 2008). Além disso, a principal vantagem da abordagem à nível de dados é que elas são mais versáteis, pois seu uso independe do classificador utilizado, dessa forma é possível realizar o pré-processamento de todos os conjuntos de dados e utilizá-los para treinar diferentes classificadores. Dessa maneira, o tempo computacional necessário para preparar os dados é requerido apenas uma vez.

Neste trabalho foram utilizados diferentes métodos de seleção de amostras em conjunto com *oversampling* e técnicas híbridas para ajustar a distribuição das classes no conjunto de dados de treinamento. Estes métodos são classificados por Ramentol et al. (2012) em três grupos:

- ***Undersampling*** - métodos que criam um subconjunto do conjunto de dados original através da eliminação de amostras da classe majoritária. Por exemplo, "Tomek links" (TOMEK, 1976) e "Neighborhood Cleaning Rule" que utiliza *Wilson's Edited Nearest Neighbor Rule* (ENN) para remover amostras da classe majoritária (WILSON, 1972).

- ***Oversampling*** - métodos que criam um novo conjunto de dados a partir do original através da replicação de algumas amostras da classe minoritária ou através da criação de novas amostras com base nas amostras da classe minoritária. Por exemplo, "Random over-sampling" ROS (BATISTA et al., 2004), "Synthetic Minority Oversampling Technique" SMOTE (CHAWLA et al., 2002), Borderline-SMOTE (HAN et al., 2005) e Safe-Level-SMOTE (BUNKHUMPORNPAT et al., 2009).

- **Híbrido** - métodos que combinam os dois métodos anteriores, eliminando algumas amostras da classe minoritária que foram expandidas pelos métodos de *oversampling* a fim de eliminar o super ajuste (*overfitting*). Por exemplo, SMOTE+ENN e SMOTE+Tomek Links, propostos por (BATISTA et al., 2004).

No restante desse capítulo serão apresentados os métodos de pré-processamento utilizados para realizar o balanceamento dos conjuntos de dados utilizados nos experimentos desse trabalho.

4.1 ROS - RANDOM OVER-SAMPLING

Este método proposto em Batista et al. (2004) realiza o balanceamento da distribuição das classes através da replicação randômica das amostras da classe minoritária. Métodos de *oversampling* randômicos podem aumentar a probabilidade da ocorrência de *overfitting*, uma vez que ele produz cópias exatas das amostras da classe minoritária. Dessa maneira, um classificador simbólico, por exemplo, deve construir regras que são aparentemente precisas, mas na verdade funciona apenas para as amostras replicadas.

Este método foi utilizado neste trabalho como *baseline* para comparação com os demais métodos utilizados uma vez que ele realiza o balanceamento das amostras apenas duplicando as amostras da classe minoritária.

4.2 SMOTE: SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE

O método SMOTE proposto por Chawla et al. (2002) é um adas técnicas de *oversampling* mais utilizadas na literatura. Este método propõe que novas amostras sintéticas da classe minoritária sejam criadas ao invés de apenas replicar as amostras existentes. O SMOTE opera no espaço dos atributos criando novas amostras da classe minoritária a partir da junção de alguns ou todos os k vizinhos mais próximos. O Algoritmo 1 apresenta o pseudocódigo do método SMOTE.

De acordo com a quantidade de novas amostras a serem criadas, alguns dos k vizinhos mais próximos são escolhidos de forma randômica. Por padrão o SMOTE utiliza cinco vizinhos mais próximos. Por exemplo, se a quantidade de novas amostras for 200%, apenas dois dos cinco vizinhos mais próximos serão escolhidos e uma nova amostra será criada na direção de cada um. Estre processo é ilustrado na Figura 28, onde x_i é o ponto selecionado, x_{i1} até x_{i4} são alguns dos vizinhos selecionados e r_{i1} até r_{i4} são os pontos sintéticos criados pela interpolação randômica.

Amostras sintéticas são criadas da seguinte maneira: Toma-se a diferença entre o vetor de características da amostra em questão e seu vizinho mais próximo. Multiplica-se essa diferença por um valor randômico entre 0 e 1, adiciona esse valor ao vetor de características em questão. Isto provoca a seleção de um novo ponto entre a linha de segmento das duas características específicas. Esta abordagem força efetivamente que a região de decisão da classe minoritária se torne mais generalizada. A Tabela 3 apresenta um exemplo do cálculo de uma amostra sintética randômica.

Algoritmo 1: SMOTE(*Nmin, NS, k*).

Entrada: *Número de amostras da classe minoritária Nmin; Quantidade % de SMOTE NS; Número de vizinhos mais próximos k.*

Saída: $(NS \div 100) \times Nmin$ amostras sintéticas da classe minoritária

início
- *(* Se NS for menor que 100%, randomize as amostras da classe minoritária pois apenas uma porcentagem aleatória delas serão utilizadas no SMOTE. *)*
- **se** $NS < 100$ **então** Randomize os *Nmin* amostras da classe minoritária
  - $Nmin = (NS \div 100) \times Nmin$
  - $NS = 100$
- **fim**
- $NS = (int)(NS \div 100)$ *(* A quantidade de SMOTE é assumida ser em múltiplos integrais de 100. *)*
- $k =$ Número de vizinhos próximos
- $numattrs =$ Número de atributos
- *Sample*[][]: vetor para amostras originais da classe minoritária
- *newindex*: mantém a contagem do número de amostras sintéticas geradas, iniciado em 0
- *Synthetic*[][]: vetor para amostras sintéticas
- *(* Calcule k vizinhos mais próximos para cada amostra da classe minoritária apenas. *)*
- **para** *(i ← 1* ***até*** *Nmin)* **faça** Calcule *k* vizinhos mais próximos para *i*, e salve o índice no *nnarray*
  - *Populate(NS, i, nnarray)*
- **fim**
- *Populate(NS, i, nnarray) (* Função para gerar a amostra sintética. *)*
- **enquanto** *(NS* $\neq 0$*)* **faça** Escolha um número randômico entre 1 e *k*, chame-o *nn*. Este passo escolhe um dos *k* vizinhos mais próximos de *i*.
  - **para** *attr ← 1* ***até*** *numattrs* **faça**
    - Calcule: $dif =$ Sample[nnarray[nn]][attr] $-$ Sample[i][attr]
    - Calcule: *gap = número aleatório entre 0 e 1*
    - Synthetic[newindex][attr] = Sample[i][attr] $+ gap \times dif$
  - **fim**
  - $newindex++$
  - $NS = NS - 1$
- **fim**
- **retorne** (* *Fim do Populate.* *)

fim

Fonte: Adaptado de (CHAWLA et al., 2002)

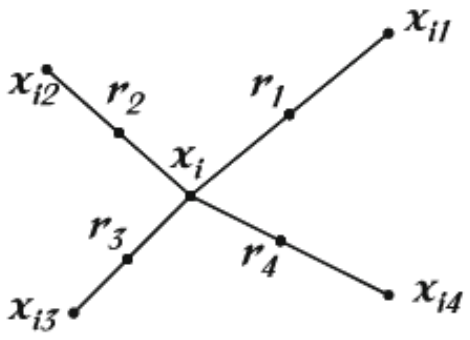


Figura 28: Uma ilustração de como são criados os pontos sintéticos no Algoritmo *SMOTE*
Fonte: (RAMENTOL et al., 2012)

Tabela 3: Exemplo da geração de amostra sintética pelo SMOTE

Considere uma amostra (6,4) e deixe (4,3) ser seu vizinho mais próximo.
(6,4) é a amostra para qual *k*-vizinhos mais próximos estão sendo identificados.
(4,3) é um dos *k*-vizinhos mais próximos.
Temos:
f1_1 = 6 f2_1 = 4, f2_1 - f1_1 = -2
f1_2 = 4 f2_2 = 3, f2_2 - f1_2 = -1
A nova amostra será gerada como
$(f1', f2') = (6,4) + rand(0-1) \times (-2,-1)$
$rand(0-1)$ gera um número aleatório entre 0 e 1.

Fonte: Adaptado de (CHAWLA et al., 2002)

Em contraste com as técnicas de replicação comum, por exemplo, o *random oversampling* na qual a barreira de decisão se torna mais específica, com o *SMOTE* o problema do super ajuste (*overfitting*) é de certo modo evitado fazendo com que a barreira de decisão da classe minoritária seja ampliada e se espalhe pelo espaço da classe majoritária, a partir da criação de novas amostras da classe minoritária para a aprendizagem (RAMENTOL et al., 2012).

4.3 SMOTE + TOMEK LINKS

De acordo com Batista et al. (2004), embora a geração de novas amostras da classe minoritária através de *oversampling* possa balancear a distribuição das classes, alguns outros problemas comumente presentes nos conjuntos de dados com distribuição de classes desbalanceadas não são resolvidos. Frequentemente, o agrupamento das classes não é bem definido uma vez que algumas amostras da classe majoritária podem estar invadindo o espaço da classe minoritária. O oposto também pode ser verdadeiro, uma vez que a interpolação das amostras

da classe minoritária podem expandir o agrupamento da classe minoritária, introduzindo amostras artificiais da classe minoritária profundamente no espaço da classe majoritária. Induzir um classificador em tal situação pode conduzir a um *overfitting*. A fim de criar agrupamentos de classes bem definidos Batista et al. (2004) propõe a aplicação do **Tomek links** no conjunto de dados de treinamento após a aplicação do *oversampling* como um método de limpeza.

Tomek links podem ser definidos como:

Dados duas amostras E_i e E_j pertencentes a classes diferentes, e $d(E_i, E_j)$ é a distância entre E_i e E_j. Um par (E_i, E_j) é chamado de *Tomek link* se não houver uma amostra E_l, tal que $d(E_i, E_l) < d(E_i, E_j)$ ou $d(E_j, E_l) < d(E_i, E_j)$. Se duas amostras formam um *Tomek link*, então uma das duas amostras é ruído ou ambas as amostras estão na fronteira dos agrupamentos. *Tomek links* podem ser utilizados como um método de *undersampling* ou como um método de limpeza. Como um método de *undersampling*, somente amostras pertencentes a classe majoritária são eliminadas, e como um método de limpeza, amostras de ambas as classes são removidas (BATISTA et al., 2004). A aplicação desse método é ilustrado na Figura 29.

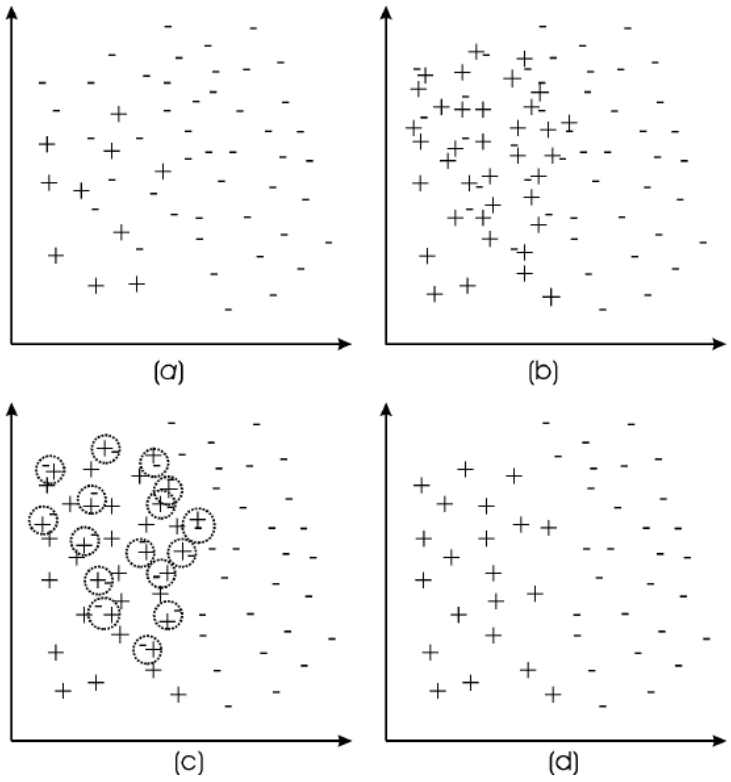


Figura 29: Balanciando um conjunto de dados: conjunto de dados original (a); conjunto de dados após *oversampling* (b); Identificação dos Tomek links (c); e remoção dos ruídos e amostras de fronteira (d)

Fonte: (BATISTA et al., 2004)

Primeiro, sobre o conjunto de dados original (a) é aplicado o método de *oversampling* SMOTE (b), e então são identificados os Tomek links (c) e removidos, produzindo um conjunto de dados balanceado com agrupamentos bem definidos (d).

Assim, ao invés de remover apenas as amostras da classe majoritária que formam *Tomek links*, amostras de ambas as classes são removidos.

4.4 SMOTE + ENN

Neighborhood Cleaning Rule (NCL) utiliza a *Wilson's Edited Nearest Neighbor Rule* (ENN) para remover amostras da classe majoritária. ENN remove qualquer amostra cujo rótulo da classe for diferente de pelo menos dois dos três vizinhos mais próximos. NCL modifica o ENN a fim de aumentar a limpeza de dados. Para um problema de duas classes o algoritmo pode ser descrito da seguinte maneira: Para cada amostra E_i no conjunto de treinamento, seus três vizinhos mais próximos são encontrados. Se E_i pertencer à classe majoritária e a classificação dada por seus três vizinhos mais próximos contradiz a classe original de E_i, então E_i é removido. Se E_i pertencer a classe minoritária e seus três vizinhos mais próximos classificarem incorretamente E_i, então os vizinhos mais próximo que pertencerem a classe majoritária são removidos do conjunto de treinamento.

A motivação por de trás desse método é similar ao Smote + Tomek links. ENN tende a remover mais amostras do que o Tomek links, então é esperado que isto irá prover uma limpeza mais profunda nos dados segundo Batista et al. (2004). O ENN é utilizado após a aplicação do SMOTE para remover amostras de ambas as classes, de modo que qualquer amostra que seja classificada de forma incorreta pelos seus três vizinhos mais próximos é removida do conjunto de dados de treinamento (BATISTA et al., 2004).

4.5 BORDERLINE-SMOTE

Um aperfeiçoamento do *SMOTE* foi proposto em Han et al. (2005), chamado de *Borderline-SMOTE*. Neste método as amostras da classe minoritária são divididas em três regiões: ruído, fronteira e seguros, levando em consideração o número de amostras da classe majoritária nos k vizinhos mais próximos. Considere *Nmaj* como o número de amostras da classe majoritária nos k vizinhos mais próximos. As três regiões são definidas através das definições contidas na Tabela 4. *Borderline-SMOTE* utiliza a mesma técnica de *oversampling* do *SMOTE*, porém ele é aplicado apenas nas amostras de fronteira da classe minoritária ao invés de todas as amostras da classe como no *SMOTE*. Considerando duas amostras da classe minoritária cujos

valores para *Nmaj* sejam iguais a *k* e *k-1* para a primeira e segunda amostra respectivamente. Essas amostras não são obviamente diferentes mas elas estão separadas em regiões diferentes, ruído e fronteira. A primeira amostra é rejeitada e a segunda amostra é selecionada para o *oversampling*.

Tabela 4: A definição de ruído, fronteira e região segura no *Borderline-SMOTE*

Região	Definição
Ruído	$Nmaj = k$
Fronteira	$\frac{1}{2}k \leq Nmaj < k$
Segura	$0 \leq Nmaj < \frac{1}{2}k$

Fonte: Adaptado de (BUNKHUMPORNPAT et al., 2009)

Com o intuito de atingir melhores predições, a maioria dos algoritmos de classificação tentam aprender os limites de cada classe da maneira mais exata possível durante o processo de treinamento. As amostras que estão na fronteira dos agrupamentos das classes e as amostras próximas à elas são mais propícias de serem incorretamente classificadas do que as amostras distantes da fronteira e assim mais importantes para a classificação.

Com base nessa análise, as amostras distantes das fronteiras tendem a contribuir pouco para a classificação e por isso no borderline-SMOTE o *oversampling* é realizado apenas nas amostras da classe minoritária que estão próximos a linha de fronteira.

O método borderline-SMOTE funciona da seguinte maneira. Primeiro, são descobertos as amostras minoritárias da fronteira; então amostras sintéticas são geradas a partir delas e adicionadas ao conjunto de dados de treinamento. Supondo que todo o conjunto de treinamento seja *CTR*, a classe minoritária seja *CMI* e a classe majoritária seja *CMA*, e

$$CMI = \{cmi_1, cmi_2, ..., cmi_{Nmin}\},\ CMA = \{cma_1, cma_2, ..., cma_{Nmaj}\}$$

onde *Nmin* e *Nmaj* são o número de amostras minoritárias e majoritárias. O procedimento detalhado do *borderline-SMOTE* é descrito como a seguir:

Passo 1. Para cada $cmi_i (i = 1, 2, ..., cmi_{Nmin})$ na classe minoritária *CMI*, são calculados os *m* vizinhos mais próximos de todo o conjunto de treinamento *CTR*. O número de amostras majoritárias dentre os *m* vizinhos mais próximos é denotado por m' $(0 \leq m' \leq m)$.

Passo 2. Se $m' = m$, todos os *m* vizinhos de cmi_i são amostras majoritárias, cmi_i é considerado ruído e não é dirigido aos próximos passos. Se $\frac{m}{2} \leq m' < m$, o número de vizinhos majoritários mais próximos de cmi_i é maior que o número de vizinhos minoritários,

cmi_i é considerado como sendo facilmente classificado incorretamente e então é colocado em um conjunto denominado *DANGER*. Se $0 \leq m' < \frac{m}{2}$, cmi_i é seguro e não é necessário prosseguir para os próximos passos.

Passo 3. As amostras no conjunto *DANGER* são os dados de fronteira da classe minoritária *CMI*, e podemos observar que $DANGER \subseteq CMI$. Temos

$$DANGER = \{cmi'_1, cmi'_2, ..., cmi'_{dnum}\}, \quad 0 \leq dnum \leq Nmin$$

Para cada amostra em *DANGER*, são calculados os k vizinhos mais próximos a partir de *CMI*.

Passo 4. Nesse passo, são gerados $s \times dnum$ amostras sintéticas da classe minoritária dos dados em *DANGER*, onde s é um inteiro entre 1 e k. Para cada cmi'_i, são selecionados aleatoriamente s vizinhos mais próximos a partir dos k vizinhos mais próximos em *CMI*. Primeiramente, são calculadas as diferenças, $dif_j (j = 1, 2, ..., s)$ entre cmi'_i e seus s vizinhos mais próximos de *CMI*, então multiplica-se a dif_j por um número aleatório $r_j (j = 1, 2, ..., s)$ entre 0 e 1, finalmente, s novas amostras sintéticas da classe minoritária são geradas entre cmi'_i e seus vizinhos mais próximos:

$$synthetic_j = cmi'_i + r_j \times dif_j, \quad j = 1, 2, ..., s$$

O procedimento acima é repetido para cada cmi'_i no DANGER e pode alcançar $s \times dnum$ amostras sintéticas. Este passo é similar ao SMOTE Original (CHAWLA et al., 2002). No procedimento acima, cmi_i, cma_i, cmi'_i, dif_j e $synthetic_j$ são vetores. Os dados sintéticos são gerados ao longo da linha das amostras de fronteira da classe minoritária e seus vizinhos mais próximos da mesma classe, desse modo a fronteira da classe é fortalecida.

Este método pode ser facilmente compreendido através do seguinte conjunto de dados simulado, *Circle*, que possui duas classes. A Figura 30 (a) mostra a distribuição original do conjunto de dados, os pontos circulados representam as amostras majoritárias e os sinais positivos são amostras minoritárias. Primeiramente, é aplicado o *borderline-SMOTE* para encontrar as amostras de fronteira da classe minoritária, que são denotados por quadrados sólidos na Figura 30 (b). Então, novas amostras sintéticas são geradas através das amostras de fronteira da classe minoritária. As amostras sintéticas são mostradas na Figura 30 (c) com quadrados vazios. Através da Figura 30 fica fácil identificar que diferentemente do SMOTE, esse método realiza *oversampling* da fronteira e seus pontos mais próximos da classe minoritária.

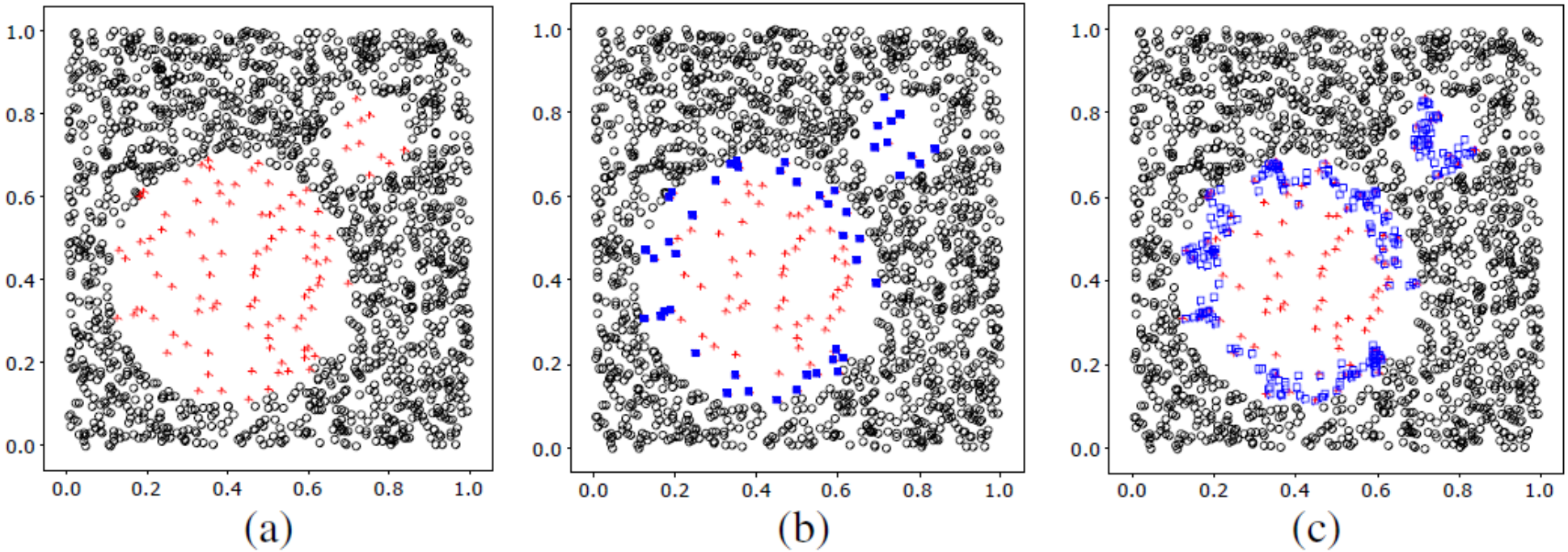


Figura 30: (a) Distribuição original do conjunto de dados Circle. (b) As amostras de fronteira da classe minoritária (quadrados sólidos). (c) As amostras de fronteira sintéticas (quadrados vazios)

Fonte: (HAN et al., 2005)

4.6 SAFE-LEVEL-SMOTE

As amostras sintéticas criadas pelo SMOTE (CHAWLA et al., 2002) fazem com que o classificador crie uma região de decisão maior e menos específica aumentando o poder de generalização do classificador. Contudo, segundo Bunkhumpornpat et al. (2009), o SMOTE enfrenta o problema de super generalização em função de generalizar de maneira cega a região da classe minoritária sem considerar a classe majoritária. Esta estratégia é particularmente problemática no caso de distribuições de classes altamente enviesadas, nesses casos, uma classe minoritária é muito esparsa em relação a classe majoritária, resultando assim em uma grande chance de que as classes fiquem misturadas.

Proposto por Bunkhumpornpat et al. (2009) e baseado no SMOTE, o Safe-Level-SMOTE atribuí a cada instância da classe minoritária um nível de segurança denominado *safe level* antes de gerar instâncias sintéticas. Cada instância sintética é posicionada o mais próximo possível do maior nível de segurança de modo que todas as instâncias sintéticas sejam geradas apenas em regiões seguras.

O *safe level (sl)* é definido pela Equação 20. Se o *safe level* de uma amostra está próximo de 0, a amostra está próxima de ruído. Se está próximo de k, a amostra é considerada segura. O *safe level ratio* é definido pela Equação 21. Este é utilizado para selecionar uma posição segura para a geração sintética das amostras.

$$\textit{safe level (sl)} = \textit{o número de amostras da classe minoritária contido em k vizinhos mais próximos.} \tag{20}$$

$$\textit{safe level ratio} = \textit{sl de uma amostra da classe minoritária} \div \textit{sl dos vizinhos mais próximos.} \tag{21}$$

O algoritmo *Safe-Level-SMOTE* é apresentado no Algoritmo 2. Todas as variáveis do algoritmo são descritas a seguir: *cmi* é uma amostra no conjunto de todas as amostras originais da classe minoritária em *CMI*; *v* é um vizinho mais próximo selecionado de *cmi*; f incluído no conjunto de todas as amostras sintéticas da classe minoritária CMI' é uma amostra sintética; sl_{cmi} e sl_v são *safe level* de *cmi* e *safe level* de *v* respectivamente; *sl_ratio* é o *safe level ratio*; *numattrs* é o número de atributos; *dif* é a diferença entre os valores dos atributos de *v* e *cmi*; *gap* é uma fração aleatória de *dif*; *cmi[i]*, *v[i]* e *f[i]* são os valores numéricos de um determinado atributo da amostra na i^{a} posição; *cmi*, *v* e f são vetores; sl_{cmi}, sl_v, *sl_ratio*, *numattrs*, *dif* e *gap*

são variáveis escalares (que podem conter um único valor inteiro ou de texto).

Após atribuir o *safe level* para *cmi* e o *safe level* para *v*, o algoritmo calcula o *safe level ratio*. Existem cinco casos correspondentes para o valor do *safe level ratio* mostrados nas linhas 13 a 29 do Algoritmo 2.

O primeiro caso é apresentado nas linhas 13 a 15 do Algoritmo 2. O *safe level ratio* é igual a ∞ e o *safe level* de *cmi* é igual a 0. Isto significa que tanto *cmi* e *v* são ruídos. Se este caso ocorrer, amostras sintéticas não serão criadas porque o algoritmo não deseja enfatizar a importância de regiões ruidosas.

O segundo caso é apresentado nas linhas 18 a 20 do Algoritmo 2. O *safe level ratio* é igual a ∞ e o *safe level* de *cmi* é diferente de 0. Isto significa que *v* é ruído. Se este caso ocorrer, uma amostra sintética será gerada distante do exemplo ruidoso *v* através da duplicação de *cmi* porque o algoritmo deseja evitar a amostra ruidosa *v*.

O terceiro caso é apresentando nas linhas 21 a 23 do Algoritmo 2. O *safe level ratio* é igual a 1. Isto significa que o *safe level* de *cmi* e *v* são iguais. Se este caso ocorrer, uma amostra sintética será gerada ao longo de uma linha entre *cmi* e *v* porque *cmi* é tão seguro quanto *v*.

O quarto caso é apresentado nas linhas 24 e 26 do Algoritmo 2. O *safe level ratio* é maior que 1. Isto significa que o *safe level* de *cmi* é maior do que o de *v*. Se este caso ocorrer, uma amostra sintética é posicionada o mais próximo de *cmi* porque *cmi* é mais segura que *v*. A amostra sintética será gerada entre o intervalo [0, $1 \div$ *safe level ratio*].

O quinto caso é apresentado nas linhas 27 a 29 do Algoritmo 2. O *safe level ratio* é menor que 1. Isto significa que o *safe level* de *cmi* é menor que o de *v*. Se este caso ocorrer, uma amostra sintética é posicionada próximo a *v* porque *v* é mais seguro que *cmi*. A amostra sintética será gerada no intervalo entre [1 - *safe level ratio*, 1].

Após o término de cada interação do laço na linha 3, se o primeiro caso não correr, uma amostra sintética f será criada ao longo do intervalo específico entre *cmi* e *v*, e então f será adicionada ao conjunto de todas as amostras sintéticas da classe minoritária CMI'.

Ao final do algoritmo, é retornado um conjunto de todas as amostras sintéticas CMI'. O algoritmo gera $|CMI| - ns$ amostras sintéticas onde $|CMI|$ é o número de todas as amostras da classe minoritária em *CMI*, e *ns* é o número de amostras que satisfaça a condição do primeiro caso.

Os experimentos realizados em Bunkhumpornpat et al. (2009) mostram que a performance do *Safe-Level-SMOTE* avaliada através da Precisão e F-measure são melhores do que o

Algoritmo 2: Safe-Level-SMOTE.

Entrada: *Um conjunto de todas as amostras da classe minoritária originais CMI*
Saída: *Um conjunto de todas as amostras sintéticas da classe minoritária CMI'*

início
  CMI' = Ø
  para cada *(amostra da classe minoritária cmi em CMI)* **faça**
    calcule *k* vizinhos mais próximos para *cmi* em *CMI* e aleatoriamente selecione um dos *k* vizinhos mais próximos, chame-o de *v*.
    sl_{cmi} = o número de amostras da classe minoritária em *k* vizinhos mais próximos para *cmi* em *CMI*
    sl_v = o número de amostras da classe minoritária em *k* vizinhos mais próximos para *v* em *CMI*
    se *(*$sl_v \neq 0$*)* **então** ; *sl* é *safe level*.
      sl_ratio = $sl_{cmi} \div sl_v$; sl_ratio é taxa de safe level.
    fim
    senão
      sl_ratio = ∞
    fim
    se *(*$sl\_ratio = \infty$ *E* $sl_{cmi} = 0$ *)* **então** ; 1ºcaso.
      Não gera amostras sintéticas da classe minoritária.
    fim
    senão
      para *(atti = 1* ***até*** *numattrs)* **faça** ; numattrs é o número de atributos.
        se *(*$sl\_ratio = \infty$ *E* $sl_{cmi} \neq 0$*)* **então** ; 2ºcaso.
          gap = 0
        fim
        senão se *(sl_ratio = 1)* **então** ; 3ºcaso.
          gerar aleatoriamente um número ente 0 e 1, chame-o de gap
        fim
        senão se *(*$sl\_ratio > 1$*)* **então** ; 4ºcaso.
          gerar aleatoriamente um número ente 0 e 1 $\div$ sl_ratio, chame-o de gap
        fim
        senão se *(*$sl\_ratio < 1$*)* **então** ; 5ºcaso.
          gerar aleatoriamente um número ente 1 - sl_ratio e 1, chame-o de gap
        fim
        $dif = v[atti] - cmi[atti]$
        $f[atti] = cmi[atti] + gap \times dif$
      fim
      $CMI' = CMI' \bigcup \{f\}$
    fim
  fim
  retorne CMI'
fim

Fonte: Adaptado de (BUNKHUMPORNPAT et al., 2009)

SMOTE e *Borderline-SMOTE* quando o algoritmo de árvore de decisão C4.5 é aplicado como classificador. Isto ocorre pois o *Safe-Level-SMOTE* realiza *oversampling* de maneira mais cuidadosa no conjunto de dados. Cada amostra sintética é gerada em uma posição segura considerando o *safe level ratio* das amostras. Em contraste, *SMOTE* e *Borderline-SMOTE* podem gerar amostras sintéticas em localizações inapropriadas, tal como em regiões de sobreposições e ruídos. Os autores do método concluíram que amostras sintéticas geradas em posições seguras podem melhorar a performance de predição dos classificadores para a classe minoritária.

4.7 SPIDER

Posposto em Stefanowski e Wilk (2008), esta abordagem utiliza uma distinção entre dois tipos de amostras, *ruído* e *seguro*. Amostras seguras devem ser corretamente classificadas por um classificador, enquanto ruídos são prováveis de serem classificados incorretamente e requerem um processamento especial. Para descobrir o tipo de uma amostra é aplicado *nearest neighbor rule* (NNR) com a distância métrica de valor heterogêneo chamada *Heterogeneous Value Difference Metric* (HVDM), capaz de lidar com atributos numéricos e nominais, apresentada em Wilson e Martinez (2000). Uma amostra é segura se ela é corretamente classificada por seus k vizinhos mais próximos, caso contrário ela é *ruído*. As amostras são preprocessadas de acordo com seus tipos, e as amostras ruidosas da classe majoritária são manejados segundo os princípios do *Edited Nearest Neighbor* (ENNR) (WILSON; MARTINEZ, 2000).

Esta abordagem é apresentada em mais detalhes no Algoritmo 3. Neste algoritmo o *CMI* é utilizado para denotar a classe minoritária e o *CMA* para a classe majoritária. Também são utilizados os rótulos *seguro* e *ruído* para indicar o tipo apropriado das amostras. Além disso são utilizadas duas funções: *classify_knn(t, k)* e *knn(t, k, c, o)*. A primeira função classifica t utilizando seus k vizinhos mais próximos e retorna a informação indicando se a classificação está correta ou não. A segunda função identifica k vizinhos mais próximos de t e retorna um conjunto deles que pertencerem à classe c e são rotulados como o, sendo $c \neq o$, (o conjunto de retorno pode ser vazio se nenhum dos k vizinhos pertencerem a c ou forem rotulados como o). Por fim, é assumido que $|\,.\,|$ retorna o número de elementos de um conjunto.

A abordagem do *SPIDER* consiste em duas fases. Na primeira fase (linhas 2-9) são identificados os tipos de cada amostra através da aplicação do *NNR* e rotuladas de acordo com o resultado. Seguindo a sugestão em Laurikkala (2001), o número de vizinhos mais próximos foi ajustado para 3. Então na segunda fase (linhas 10-40) as amostras são processadas de acordo com seus rótulos. Como é desejável preservar todas as amostras de *CMI*, assume-se que apenas as amostras de *CMA* devem ser removidas (linhas 10 e 40, onde são aplicados os princípios do

Algoritmo 3: SPIDER.

Entrada: *ampl* - opção de amplificação

início
  para cada $t \in CMA \cup CMI$ **faça**
    se *classify_knn(t, 3) é correto* **então**
      rotule t como *seguro*
    fim
    senão
      rotule t como *ruído*
    fim
  fim
  D $\longleftarrow$ todos $y \in CMA$ e rotulados como *ruído*
  se *ampliação fraca* **então**
    para cada $t \in CMI$ *e rotulado como ruído* **faça**
      amplifique t criando suas $\mid knn(t,3,CMA,seguro) \mid$ cópias
    fim
  fim
  senão se *ampliação fraca e rotulação* **então**
    para cada $t \in CMI$ *e rotulado como ruído* **faça**
      amplifique t criando suas $\mid knn(t,3,CMA,seguro) \mid$ cópias
    fim
    para cada $t \in CMI$ *e rotulado como ruído* **faça**
      para cada $y \in knn(t,3,CMA,$ *ruído)* **faça**
        rotule y trocando sua classe de CMA para CMI
        $D \longleftarrow D \setminus \{y\}$
      fim
    fim
  fim
  senão {ampliação forte}
    para cada $t \in CMI$ *e rotulado como seguro* **faça**
      amplifique t criando suas $\mid knn(t,3,CMA,seguro) \mid$ cópias
    fim
    para cada $t \in CMI$ *e rotulado como ruído* **faça**
      se *classify_knn(t, 5) é correto* **então**
        amplifique t criando suas $\mid knn(t,3,CMA,seguro) \mid$ cópias
      fim
      senão
        amplifique t criando suas $\mid knn(t,5,CMA,seguro) \mid$ cópias
      fim
    fim
  fim
  remova todos $y \in D$
fim

Fonte: Adaptado de (STEFANOWSKI; WILK, 2008)

ENNR). Por outro lado, diferentemente dos outros métodos descritos anteriormente, o *SPIDER* modifica o *CMA* de maneira cuidadosa, portanto todas as amostras seguras dessa classe são preservadas, o método *neighborhood cleaning rule* (NCR) de Laurikkala (2001) é aplicado e remove alguns deles se estiverem muito próximos de amostras ruidosas de *CMI*. São propostas três diferentes técnicas para a segunda fase: *amplificação fraca*, *amplificação fraca e rotulação*, e *amplificação forte*. Todas elas envolvem modificações da classe minoritária, embora, o grau e escopo das mudanças variem.

Amplificação fraca (linhas 11-15) é a técnica mais simples. Ela é focada nas amostras ruidosas de *CMI* e os amplifica através da adição de cópias de acordo com a existência de amostras seguras de *CMA* em sua vizinhança de nível 3. Assim a amplificação é limitada as amostras difíceis de *CMI*, cercados por membros seguros de *CMA* (se não houverem vizinhos seguros, então a amostra não é amplificada). Isto aumenta o peso de tais amostras difíceis e habilita os algoritmos de aprendizado a capturá-las, enquanto elas poderiam ser descartadas como ruído caso contrário.

A segunda técnica, amplificação fraca e rotulação (linhas 16-26), também foca nas amostras ruidosas de *CMI* e estende a primeira técnica com um passo adicional de troca de rótulo. No primeiro passo (linhas 17-19) amostras ruidosas de *CMI* cercadas por amostras seguras de *CMA* são fracamente amplificadas. No passo seguinte (linhas 20-25) amostras ruidosas de *CMA* localizadas nos vizinhos mais próximos de nível 3 de amostras ruidosas de *CMI* são rotuladas novamente através da mudança de classe atribuída de *CMA* para *CMI*. Assim, é expandida a cobertura ao redor da amostra ruidosa selecionada de *CMI*, o que futuramente aumenta a sua chance de ser capturada por um classificador. Tal aumento de densidade é similar à técnica empregada pelo *SMOTE*, embora, ao invés de introduzir novas amostras artificiais são utilizadas amostras com rótulos trocados de *CMA*.

A terceira técnica, amplificação forte (linhas 27-39), é a mais sofisticada. Ela é focada em todas as amostras de *CMI* - seguras e ruidosas. Primeiro, ela processa as amostras seguras de *CMI* e as amplificam através da adição de tantas cópias quanto existirem amostras seguras de *CMA* nos 3 vizinhos mais próximos (28-29). Então ele muda para as amostras ruidosas de *CMI* (linhas 31-38). Cada uma das amostras é reclassificada utilizando a vizinhança estendida (5 vizinhos mais próximos). Se uma amostra é reclassificada corretamente, ela é amplificada de acordo com seus vizinhos (adicionando tantas cópias quanto existirem amostras seguras de *CMA* nos 3 vizinhos mais próximos), tornando-se assim suficiente para formar um *forte* padrão de classificação. Embora, se uma amostra é reclassificada incorretamente, sua amplificação é forte e o número de cópias é igual ao número de amostras seguras de *CMA* nos 5 vizinhos mais

próximos. Uma intervenção mais agressiva é causada pelo limitado número de amostras de *CMI* na vizinhança estendida e isto é necessário para fortalecer um padrão de classificação.

4.8 SPIDER2

Outro método de *re-sampling*, o SPIDER2, foi proposto em Napierala et al. (2010) e seu pseudocódigo é apresentando no Algoritmo 4. Para simplificar a notação o algoritmo não distingue entre amostras de ruído e fronteira e se refere a elas simplesmente como não-seguras.

No pseudocódigo são utilizadas as seguintes funções auxiliares:

- correct(*DS*, *t*, *k*) - classifica uma amostra *t* utilizando seus *k* vizinhos mais próximos no conjunto de dados *DS* e retorna verdadeiro ou falso para correta e incorreta classificação respectivamente;
- class(*DS*, *c*) - retorna um conjunto de amostras de *DS* que pertencerem a classe *c*;
- flagged(*DS*, *c*, *o*) - retorna um conjunto de amostras de *S* que pertencerem a classe *c* e são rotulados como *o*;
- knn(*DS*, *t*, *k*, *c*) - identifica e retorna as amostras entre os *k* vizinhos mais próximos de *t* em *DS* que pertencerem a classe *c*;
- amplify(*DS*, *t*, *k*) - amplifica amostra *t* através da criação de suas *z* cópias e adiciona-as ao *DS*. *z* é calculado como $|knn(DS,t,k,CMA)| - |knn(DS,t,k,CMI)| + 1$. Para calcular os vizinhos mais próximos é utilizado a função de distância métrica heterogênea HVDM.

SPIDER2 consiste em duas fases correspondentes ao pré-processamento do c_{maj} e c_{min} respectivamente. Na primeira fase (linhas 3-20) ele identifica as características das amostras de c_{maj}, e dependendo da opção de troca de rótulo ele pode remover ou trocar o rótulo das amostras ruidosas de c_{maj} (trocando sua classificação para c_{min}). Na segunda fase linhas (21-44) ele identifica as características das amostras de c_{min} considerando as mudanças introduzidas na primeira fase. Então, amostras ruidosas de c_{min} são amplificadas (através da replicação delas) de acordo com a opção *ampl*. Esta estrutura de duas fases é a principal diferença da primeira versão do *SPIDER*, que primeiro identifica a natureza das amostras e então processa simultaneamente c_{maj} e c_{min}. Segundo Napierala et al. (2010), tal processamento pode resultar em modificações extensivas de algumas regiões de c_{maj} e deteriorar a especificidade, este problema foi tratado no *SPIDER2*. Outras diferenças menores incluem o escopo da troca de rótulo das amostras ruidosas de c_{maj} e o nível de amplificação das amostras ruidosas de c_{min}.

Algoritmo 4: SPIDER2.

Entrada: *DS* - conjunto de dados; *CMI* - a classe minoritária; *k* - o número dos vizinhos mais próximos; *relabel* - opção de troca de rótulo (sim, não); *ampl* - opção de amplificação (não, fraca, forte).

Saída: *DS* processado.

```
1  início
2      CMA := uma classe artificial combinando todas as classes exceto a CMI
3      para cada t ∈ class(DS,CMA) faça
4          se correct(DS, t, k) então
5              rotule t como seguro
6          fim
7          senão
8              rotule t como não-segura
9          fim
10     fim
11     RS := flagged(DS, CMA, não-segura)
12     se relabel então
13         para cada y ∈ RS faça
14             mude a classificação de y para CMI
15             SR := SR \ {y}
16         fim
17     fim
18     senão
19         DS := DS \ RS
20     fim
21     para cada t ∈ class(DS,CMI) faça
22         se correct(DS, t, k) então
23             rotule t como seguro
24         fim
25         senão
26             rotule t como não-segura
27         fim
28     fim
29     se ampl = fraca então
30         para cada t ∈ flagged(DS,CMI, não-segura) faça
31             amplify(DS, t, k)
32         fim
33     fim
34     senão se ampl = forte então
35         para cada t ∈ flagged(DS,CMI, não-segura) faça
36             se correct(DS, t, k + 2) então
37                 amplify(DS, t, k)
38             fim
39             senão
40                 amplify(DS, t, k + 2)
41             fim
42         fim
43     fim
44 fim
```

Fonte: Adaptado de (NAPIERALA et al., 2010)

4.9 KEEL

O KEEL[1] (*Knowledge Extraction based on Evolutionary Learning*) é uma ferramenta JAVA de código aberta (*Open Source* GPLv3) que pode ser utilizada para um grande número de tarefas de descoberta de conhecimento em dados. Essa ferramenta provê uma interface gráfica simples que permite projetar experimentos com diferentes conjuntos de dados e algoritmos de inteligência computacional. Ela contém uma grande variedade de algoritmos clássicos de extração de conhecimento, técnicas de pré-processamento, análises, entre outras funções (ALCALA-FDEZ et al., 2009), (ALCALA-FDEZ et al., 2011).

Neste trabalho a ferramenta KEEL foi utilizada para realização das etapas de pré-processamento dos dados através dos algoritmos de balanceamento de amostras implementados pela ferramenta.

[1] http://www.keel.es.

5 METODOLOGIA

Conforme apresentado no Capítulo 3, o processo do KDD possui 5 etapas. Neste capítulo é apresentado como cada etapa foi realizada.

Para desenvolvimento deste trabalho, foram utilizados dados reais coletados pelo sistema de Informatização da Rede de Serviços da Assistência Social (IRSAS) em funcionamento no município de Cascavel/PR.

Como exemplo de volumetria, atualmente o sistema implantado no município de Cascavel/PR conta com 132.675 pessoas cadastradas e mais de 1.175.000 registros de atendimentos, denominados no sistema como ocorrências.

Como detalhado no Capítulo 2, no IRSAS são coletadas diversas informações das pessoas atendidas pelas unidades da rede de serviço da assistência social. E entre essas informações, estão os dados pessoais, logradouro, renda, composição familiar, registro de todos os atendimentos realizados pela rede de serviços da assistência social, entre outras.

Utilizou-se o processo do KDD, detalhado anteriormente na Seção 3.1, como forma de transformar os dados brutos das famílias existentes na base de dados em informações que possam apoiar o processo da Busca Ativa.

A definição do problema a ser resolvido evoluiu durante a etapa de revisão bibliográfica e a realização dos experimentos. Chegou-se a conclusão de que o sistema IRSAS contém uma grande quantidade de informações do público alvo da assistência social, que na maioria das vezes estão em situação de vulnerabilidade e risco social, que podem ser utilizados para apoiar o processo da Busca Ativa.

Desse modo definiu-se que a meta a ser alcançada através do processo de descoberta de conhecimento (KDD) é realizar a classificação de vulnerabilidade e risco social de 100% das famílias cadastradas no IRSAS, rotulando-as em classes de Baixa, Média ou Alta Vulnerabilidade para apoiar o processo da Busca Ativa.

5.1 SELEÇÃO

A seleção dos dados apropriados para a análise foi realizada com base no objetivo a ser alcançado baseando-se no formulário de avaliação de vulnerabilidade e risco social existente e no conhecimento do especialista no domínio apresentado na Seção 2.2.

Conforme apresentado na Seção 2.2, o sistema IRSAS possui um formulário de avaliação de vulnerabilidade e risco social onde é possível obter um índice de vulnerabilidade e risco social da família. Porém, tendo em vista que o objetivo final é alcançar esse índice de vulnerabilidade e risco social para 100% das famílias cadastradas no sistema, foram selecionados atributos que fossem comuns a todas as famílias, sendo esses atributos: comuns a todos os membros da família, atributos do responsável familiar e atributos de todos os dependentes.

A maioria das informações está presente apenas no cadastro dos membros da família, mas algumas delas também são utilizadas no formulário de avaliação de vulnerabilidade e risco social.

Na prática foram selecionadas as informações de preenchimento obrigatório durante a inserção da família no IRSAS e alguns atributos foram calculados com base em outras informações disponíveis, como por exemplo, os atributos "FAMILIA_MONOPARENTAL, RENDA_PERCAPTA, RENDA_TOTAL, etc.".

A Tabela 5 apresenta os atributos utilizados. Os atributos referentes ao DEPENDENTE 1 são repetidos de acordo com a quantidade de dependentes de cada família.

Os atributos com ID 1 a 4 referem-se a informação de todos os membros familiares. Os atributos de 5 a 19 referem-se ao responsável da família que normalmente é o pai ou a mãe. Os atributos 20 a 33 referem-se a um membro da família que não é o responsável, neste caso ele é chamado de dependente. Então se a família tiver 4 dependentes os atributos 20 ao 33 serão repetidos 4 vezes. Nestes experimentos a maior família encontrada possuía 1 responsável e 14 dependentes.

As informações utilizadas estão dispersas em diversas tabelas no banco de dados do sistema e para seleção desses atributos foi construída uma consulta em Linguagem de Consulta Estruturada, em inglês *Structured Query Language* (SQL), para extrair os dados da base de dados do IRSAS. A Figura 31 apresenta a relação das tabelas e seus relacionamentos.

Tabela 5: Atributos selecionados

ID	COLUNA	DESCRIÇÃO	TIPO
01	NUMERO_PESSOAS	Numero de pessoas na família	Numérico
02	RENDA_TOTAL	Renda Total Familiar	Numérico
03	RENDA_PERCAPTA	Renda Per Capta Familiar	Numérico
04	FAMILIA_MONOPARENTAL	Família possui apenas um responsável	Numérico
05	ID_PESSOA_RESP	Código do responsável	Numérico
06	IDADE_RESP	Idade Responsável	Numérico
07	RENDA_RESP	Renda Responsável	Numérico
08	ESCOLARIDADE_RESP	Escolaridade Responsável	Categórico
09	TIPO_HABITACAO_RESP	Tipo de habitação do responsável	Categórico
10	TIPO_LOGRADOURO_RESP	Tipo de logradouro do responsável	Categórico
11	ENDERECO_RESP	Endereço do responsável	Categórico
12	OCUPACAO_RESP	Ocupação Responsável	Categórico
13	DEFICIENCIA_RESP	Deficiência Responsável	Categórico
14	RACA_RESP	Raça Responsável	Categórico
15	SEXO_RESP	Sexo do responsável	Categórico
16	ORIENTACAO_SEXUAL_RESP	Orientação Sexual Responsável	Categórico
17	PARENTESCO_RESP	Parentesco Responsável	Categórico
18	DESEMPREGADO	Responsável está desempregado	Numérico
19	ATEND_ULTIMO_6MESES_RESP	Se responsável foi atendido nos últimos 6 meses	Numérico
20	ENT_ULTIMO_ATEND_RESP	Unidade último atendimento Responsável	Categórico
21	ID_DEPENDENTE_1	Código do dependente 1	Numérico
22	IDADE_DEPENDENTE_1	Idade Dependente 1	Numérico
23	RENDA_DEPENDENTE_1	Renda Dependente 1	Numérico
24	ESCOLARIDADE_DEPENDENTE_1	Escolaridade Dependente 1	Categórico
25	TIPO_HABITACAO_DEPENDENTE_1	Tipo de habitação Dependente 1	Categórico
26	TIPO_LOGRADOURO_DEPENDENTE_1	Tipo de logradouro Dependente 1	Categórico
27	ENDERECO_DEPENDENTE_1	Endereço do Dependente 1	Categórico
28	OCUPACAO_DEPENDENTE_1	Ocupação Dependente 1	Categórico
29	DEFICIENCIA_DEPENDENTE_1	Deficiência Dependente 1	Categórico
30	RACA_DEPENDENTE_1	Raça Dependente 1	Categórico
31	ORIENTACAO_SEXUAL_DEPENDENTE_1	Orientação Sexual Dependente 1	Categórico
32	PARENTESCO_DEPENDENTE_1	Parentesco Dependente 1	Categórico
33	ATEND_ULTIMO_6MESES_DEPENDENTE_1	Se dependente foi atendido nos últimos 6 meses	Numérico
34	ENT_ULTIMO_ATEND_DEPENDENTE_1	Unidade último atendimento Dependente 1	Categórico

Fonte: Autoria própria.

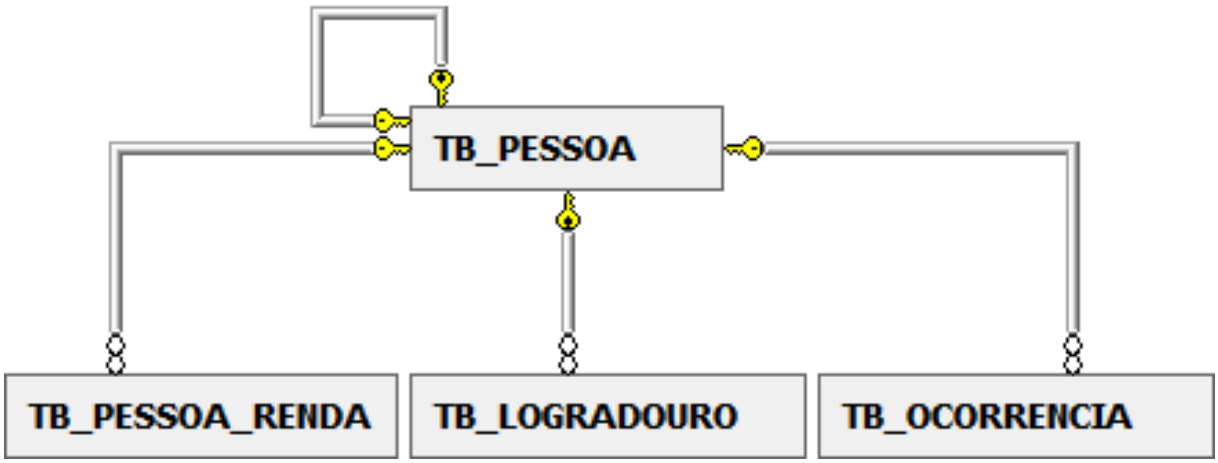


Figura 31: Relação de tabelas originárias dos atributos utilizados e seus relacionamentos

Fonte: Autoria própria.

5.2 PRÉ-PROCESSAMENTO

Nessa etapa foi realizada a eliminação de ruídos e erros através da elaboração de procedimentos para verificação da falta de dados.

Para a construção de um conjunto de dados consistente foi necessário identificar a falta de dados e substituí-los por informações que não afetassem o resultado dos algoritmos de mineração de dados.

Para realização dessa etapa foi construído um procedimento em linguagem SQL para substituir os valores faltantes dos atributos numéricos para -1 e os valores faltantes dos atributos categóricos para N/A. Esse procedimento também foi necessário para que não ocorressem erros durante a leitura do conjunto de dados pela ferramenta WEKA. Em função do número de dependentes variar de 0 até 14, fez-se necessário realizar uma consulta que retornasse as informações de 14 dependentes para todas as famílias. No caso das famílias que possuem menos de 14 dependentes, esses dados não eram retornados pela consulta gerando uma situação de falta de dados.

A Figura 32 apresenta uma parte do conjunto de dados extraído do sistema IRSAS com dados faltantes e a Figura 33 apresenta a mesma parte do conjunto após o pré-processamento realizado.

	R	S	T	U	V
1	ID_DEPENDENTE_1	IDADE_DEPENDENTE_1	RENDA_DEPENDENTE_1	ESCOLARIDADE_DEPENDENTE_1	TIPO_HABITACAO_DEPENDENTE_1
2	27106	48	0.00	Séries Finais do Ensino Fundamental Incompleto	
3					
4					
5	4458	14	0.00	Séries Finais do Ensino Fundamental Incompleto	
6	27902	14	0.00	Séries Finais do Ensino Fundamental Incompleto	OCUPAÇÃO
7	28239	9	0.00	Séries Finais do Ensino Fundamental Incompleto	PRÓPRIA
8	1480	16	0.00	Séries Finais do Ensino Fundamental Incompleto	
9	28404	16	0.00	Ensino Médio Incompleto	
10	106456	30	0.00	Séries Iniciais do Ensino Fundamental Incompleto	PRÓPRIA
11	29624	65	622.00	Ensino Médio Incompleto	
12					
13	20215	88	1167.00	Séries Iniciais do Ensino Fundamental Incompleto	NÃO INFORMADO

Figura 32: Parte do conjunto de dados extraído do sistema IRSAS com dados faltantes
Fonte: Autoria própria.

	R	S	T	U	V
1	ID_DEPENDENTE_1	IDADE_DEPENDENTE_1	RENDA_DEPENDENTE_1	ESCOLARIDADE_DEPENDENTE_1	TIPO_HABITACAO_DEPENDENTE_1
2	27106	48	0.00	Séries Finais do Ensino Fundamental Incompleto	N/A
3	-1	-1	-1.00	N/A	N/A
4	-1	-1	-1.00	N/A	N/A
5	4458	14	0.00	Séries Finais do Ensino Fundamental Incompleto	N/A
6	27902	14	0.00	Séries Finais do Ensino Fundamental Incompleto	OCUPAÇÃO
7	28239	9	0.00	Séries Finais do Ensino Fundamental Incompleto	PRÓPRIA
8	1480	16	0.00	Séries Finais do Ensino Fundamental Incompleto	N/A
9	28404	16	0.00	Ensino Médio Incompleto	N/A
10	106456	30	0.00	Séries Iniciais do Ensino Fundamental Incompleto	PRÓPRIA
11	29624	65	622.00	Ensino Médio Incompleto	N/A
12	-1	-1	-1.00	N/A	N/A
13	20215	88	1167.00	Séries Iniciais do Ensino Fundamental Incompleto	NÃO INFORMADO

Figura 33: Parte do conjunto de dados extraído do sistema IRSAS após transformação de dados faltantes

Fonte: Autoria própria.

5.3 FORMATAÇÃO

A etapa de formatação foi realizada diretamente em cada um dos experimentos. Para realização dos experimentos realizados foram selecionados atributos específicos a fim de averiguar se havia melhora nos resultados obtidos. Nos capítulos 6 e 7 onde são descritos os experimentos realizados, são apresentados quais atributos foram selecionados em cada experimento e também é explicado o motivo pelo qual tais atributos foram selecionados. Durante os experimentos foram utilizados atributos do cadastro da família, atributos da avaliação de vulnerabilidade e risco social do IRSAS existente no município de Cascavel/PR e atributos das duas fontes combinados.

5.4 MINERAÇÃO DE DADOS

Na etapa de mineração de dados, foi realizada a aplicação de dois algoritmos de classificação supervisionados (*Naive Bayes* e *AODE*) para descoberta de padrões nos dados reais obtidos da base de dados do IRSAS.

Foram utilizados os classificadores Bayesianos em função de gerarem uma estrutura de interdependência entre os atributos que pode ser útil para avaliar a situação de vulnerabilidade e risco social pelas assistentes sociais, uma vez que esses métodos tornam mais fácil para os usuários compreender a lógica do processo de classificação.

Além disto, os classificadores Bayesianos podem ser utilizados para prever qual a probabilidade de uma determinada família estar em situação de vulnerabilidade e risco social o que auxiliaria no processo de busca ativa mesmo se o classificador atingir uma alta acurácia na

rotulação das amostras.

Para realização dos experimentos foi utilizada a ferramenta WEKA que implementa diversos algoritmos de mineração de dados. Os algoritmos utilizados nesse trabalho estão implementados no WEKA com os seguintes nomes: *Naive Bayes* e *AODE* respectivamente.

Para que o WEKA possa fazer a leitura dos dados os mesmos precisam estar em um formato específico chamado *Attribute-Relation File Format* (ARFF). Por isso foi desenvolvido um algoritmo que transforma os dados consultados da base de dados objeto-relacional em um arquivo texto no formato esperado pela ferramenta. Na Figura 34 é apresentado parte do arquivo gerado antes da discretização.

```
@RELATION 'vulnerabilidade'
@ATTRIBUTE NUMERO_PESSOAS {'8','1','7','6','5','11','3','2','4','9'}
@ATTRIBUTE RENDA_TOTAL {'1436.00','995.00','0.00','650.00','800.00','300.00','922.00','1000.00','750.0
@ATTRIBUTE RENDA_PERCAPTA {'179.50','124.38','0.00','92.86','133.33','50.00','153.67','166.67','125.00
@ATTRIBUTE ID_PESSOA_RESP {'63653','28240','79284','20012','150','375','70431','60353','94033','73697'
@ATTRIBUTE IDADE_RESP {'48','27','20','37','40','62','53','44','25','34','29','36','30','71','23','24'
@ATTRIBUTE RENDA_RESP {'0.00','200.00','650.00','750.00','600.00','700.00','545.00','400.00','240.00',
@ATTRIBUTE ESCOLARIDADE_RESP {'Séries Iniciais do Ensino Fundamental Incompleto','Ensino Fundamental C
@ATTRIBUTE TIPO_HABITACAO_RESP {'CEDIDA','ALUGUEL','N/A','PRÓPRIA','FINANCIADA','NÃO INFORMADO','OUTRO
@ATTRIBUTE TIPO_LOGRADOURO_RESP {'DOMICÍLIO','ACOLHIMENTO','OUTROS','MORA COM RESPONSÁVEL','N/A'}
@ATTRIBUTE OCUPACAO_RESP {'Trabalho Informal Regular','Não informado','Dona de casa','Trabalho Informa
@ATTRIBUTE DEFICIENCIA_RESP {'Não possui','Deficiência visual','Deficiência física','Deficiência intel
@ATTRIBUTE RACA_RESP {'Branca','Parda','Negra','Não informado'}
@ATTRIBUTE ORIENTACAO_SEXUAL_RESP {'N/A','Heterosexual'}
@ATTRIBUTE PARENTESCO_RESP {'Mãe/Responsável','Não informado','Esposo(a)','Filho(a)','Companheira(o)'}
```

Figura 34: Parte do arquivo ARFF gerado para ser utilizado no WEKA

Fonte: Autoria própria.

5.5 INTERPRETAÇÃO/AVALIAÇÃO

Na etapa de interpretação e avaliação dos dados é utilizada a taxa de acerto e a matriz de confusão para avaliar o grau de assertividade de cada algoritmo utilizado.

Com base nos rótulos das classes existentes em cada uma das amostras do conjunto de treinamento, a ferramenta WEKA gera ao final do processamento a taxa de acerto e erro e a matriz de confusão.

A taxa de acerto representa em porcentagem a quantidade de amostras que foram classificadas corretamente, enquanto que a taxa de erro representa em porcentagem a quantidade de amostras classificadas incorretamente.

A matriz de confusão apresenta um resultado mais detalhado onde é possível observar a quantidade de amostras classificadas corretamente e incorretamente em cada uma das classes.

6 EXPERIMENTOS PRELIMINARES

Foram realizados alguns experimentos preliminares a fim de averiguar possíveis soluções para o problema proposto. Esses resultados preliminares foram publicados no artigo (TERRIN et al., 2014) apresentado no XLI Seminário Integrado de Software e Hardware (SEMISH).

6.1 CONFIGURAÇÃO EXPERIMENTAL

Para realização dos experimentos foram gerados alguns conjuntos de dados, a partir das informações das famílias do município de Cascavel/PR existentes na base de dados do sistema IRSAS e aplicados alguns algoritmos de classificação na busca de solucionar o problema proposto.

Como o propósito é identificar o nível de vulnerabilidade de 100% das famílias cadastradas, foram escolhidos atributos comuns a todas as famílias cadastradas e descartadas as informações exclusivamente oriundas do formulário de avaliação de vulnerabilidade e risco social, já que apenas uma pequena parte das famílias (0,8%) possui essas informações coletadas, conforme descrito no Capítulo 2. Esse conjunto de atributos será referido como "atributos comuns das famílias" no restante deste trabalho.

Nas subseções seguintes são apresentados os experimentos realizados e descritos os seus resultados.

Tabela 6: Relação de atributos descritos na Tabela 5 utilizados nos experimentos preliminares

Experimento	Conjunto	Códigos dos atributos
01	Dataset_01	[01, 02, 03, 05, 06, 07, 08, 09, 10, 11, 12, 13 14, 16, 17, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34]
02	Dataset_02	[01, 02, 03, 06, 07, 08, 09, 10, 12, 13, 14]
03	Dataset_03	[03, 04, 06, 08, 09, 10, 12, 13, 14, 15, 18]
04	Dataset_04	Todos

Fonte: Autoria própria.

O problema a ser resolvido nos experimentos 1 ao 4 foi a classificação das amostras em 3 (três) classes “Baixa”, “Média” e “Alta” vulnerabilidade distribuídas conforme a Tabela 7.

Nestes experimentos foram utilizados algoritmos de classificação supervisionados (Naive Bayes e AODE) com validação cruzada fator 10.

Tabela 7: Distribuição das amostras utilizadas nos experimentos 1 ao 4

Classificação	Quantidade	%
Alta vulnerabilidade	36	10,34
Media vulnerabilidade	306	87,93
Baixa vulnerabilidade	06	1,72

Fonte: Autoria própria.

6.2 EXPERIMENTO 1 - UTILIZANDO TODOS OS ATRIBUTOS DA FAMÍLIA

No primeiro experimento foram utilizados diversos atributos da família (256 atributos), sendo esses: atributos comuns de todos os membros da família, atributos do responsável familiar e atributos de todos os dependentes.

Os atributos selecionados para o experimento podem ser observados na linha 1 da Tabela 6.

A motivação para realização desse experimento foi gerar a classificação de vulnerabilidade e risco social para 100% das famílias cadastradas no sistema utilizando apenas os atributos comuns a todas as famílias, para auxiliar na busca ativa.

A análise dos resultados da Tabela 8 mostra que os algoritmos obtiveram um suposto bom desempenho, visto que os mesmos tiveram altas taxas de acertos. Porém esses resultados não são bons, visto que ao analisar as matrizes de confusão geradas pelos algoritmos (Figura 35), pode-se perceber que os mesmos estão sempre prevendo a classe majoritária, resultando em um valor baixo da medida AUC.

Tabela 8: Resultados do experimento 1 utilizando validação cruzada fator 10

Naive Bayes			AODE		
Acerto	**Erro**	**AUC**	**Acerto**	**Erro**	**AUC**
87.931%	12.069%	0.636%	87.931%	12.069%	0.627%

Fonte: Autoria própria.

Matriz de Confusão – Exp. 1 - Naive Bayes				
a	b	C	←	Classificado como
302	0	4	|	a = Média Vulnerabilidade
6	0	0	|	b = Baixa Vulnerabilidade
32	0	4	|	c = Alta Vulnerabilidade

Matriz de Confusão – Exp. 1 - AODE				
a	b	C	←	Classificado como
306	0	0	|	a = Média Vulnerabilidade
6	0	0	|	b = Baixa Vulnerabilidade
36	0	0	|	c = Alta Vulnerabilidade

Figura 35: Matrizes de confusão do experimento 1

Fonte: Autoria própria.

Dessa forma, neste primeiro experimento pode-se perceber que utilizar apenas os atributos comuns a todas as famílias não foi suficiente para resolver o problema em questão. Uma razão para os resultados ruins obtidos pode ser que o número de atributos disponíveis das famílias não foi suficiente para treinar os algoritmos de classificação.

6.3 EXPERIMENTO 2 - UTILIZANDO APENAS OS ATRIBUTOS “CHAVES” DEFINIDOS POR UM ESPECIALISTA DO DOMÍNIO

Este segundo experimento foi realizado com a finalidade de eliminar possíveis atributos redundantes no conjunto dados em relação ao experimento anterior descrito na Seção 6.2.

A seleção dos atributos que compõem o conjunto de dados foi realizada com base no conhecimento prévio do autor desse trabalho que participou de algumas das discussões para criação do formulário de avaliação de vulnerabilidade e risco social existente no IRSAS. Nestas reuniões o autor pôde constatar quais atributos se mostram na prática, mais relevantes para classificação da situação de vulnerabilidade das famílias. Foram selecionados apenas os atributos “chaves” da família (11 atributos) que podem ser observados na linha 2 da Tabela 6.

Os resultados do experimento 2 são apresentados na Tabela 9.

Tabela 9: Resultados do experimento 2 utilizando validação cruzada fator 10

Naive Bayes			AODE		
Acerto	**Erro**	**AUC**	**Acerto**	**Erro**	**AUC**
86.206%	13.793%	0.634%	86.494%	13.505%	0.636%

Fonte: Autoria própria.

Com a utilização desse conjunto de dados contendo apenas os atributos “chaves” da família, os resultados foram bastante similares ao experimento anterior descrito na Seção 6.2. Em uma primeira análise os mesmos também apresentaram taxas de acertos consideravelmente boas como pode ser observado na Tabela 9. Porém, ao analisar as matrizes de confusão dos classificadores (Figura 36), pode-se constatar que novamente a alta taxa de acerto está atrelada

Matriz de Confusão – Exp. 2 - Naive Bayes				
a	b	C	←	Classificado como
298	1	7	|	a = Média Vulnerabilidade
6	0	0	|	b = Baixa Vulnerabilidade
32	2	2	|	c = Alta Vulnerabilidade

Matriz de Confusão – Exp. 2 - AODE				
a	b	C	←	Classificado como
300	1	5	|	a = Média Vulnerabilidade
6	0	0	|	b = Baixa Vulnerabilidade
35	0	1	|	c = Alta Vulnerabilidade

Figura 36: Matrizes de confusão do experimento 2

Fonte: Autoria própria.

ao fato de todos os classificadores sempre preverem a classe majoritária, resultando em um valor baixo da medida AUC.

Com este segundo experimento concluiu-se que mesmo reduzindo a quantidade de atributos no conjunto de dados os algoritmos utilizados mantiveram suas taxas de acerto, porém os resultados continuaram não sendo bons em função dos modelos preverem quase sempre a classe majoritária. De uma maneira geral, a mudança realizada no conjunto de dados não alterou os resultados quando comparados ao experimento descrito na Seção 6.2.

6.4 EXPERIMENTO 3 - USANDO APENAS OS ATRIBUTOS DO FORMULÁRIO DE AVALIAÇÃO DE VULNERABILIDADE

Neste terceiro experimento realizou-se uma modificação no conjunto de dados com a finalidade de verificar se utilizando apenas os atributos que são comuns para todas as famílias e ao mesmo tempo também foram utilizados no formulário de avaliação de vulnerabilidade e risco social (12 atributos), seria possível realizar a classificação para 100% das famílias.

Os atributos selecionados para compor esse conjunto de dados são aqueles que são comuns para todas as famílias existentes na base de dados da aplicação e ao mesmo tempo também foram utilizados no formulário de avaliação de vulnerabilidade e risco social e podem ser vistos na linha 3 da Tabela 6.

Utilizando o conjunto de dados com atributos contidos no formulário de avaliação de vulnerabilidade e risco social, obteve-se uma melhora na taxa de acerto para o classificador AODE em relação ao experimento 2 descrito na Seção 6.3. Entretanto, mesmo com um aumento na taxa de acerto os resultados (Tabela 10) obtidos não foram bons porque os modelos continuaram a prever quase sempre a classe majoritária como pode ser observado na Figura 37.

Novamente os resultados obtidos se mostraram aquém do esperado visto que mesmo utilizando informações do formulário de avaliação de vulnerabilidade (que possui informações específicas sobre as famílias) não foram suficientes para treinar os classificadores.

Tabela 10: Resultado do experimento 3 utilizando validação cruzada fator 10

Naive Bayes			AODE		
Acerto	**Erro**	**AUC**	**Acerto**	**Erro**	**AUC**
85.919%	14.080%	0.625%	86.781%	13.218%	0.633%

Fonte: Autoria própria.

Matriz de Confusão – Exp. 3 - Naive Bayes				
a	b	C	←	Classificado como
297	2	7	|	a = Média Vulnerabilidade
6	0	0	|	b = Baixa Vulnerabilidade
32	2	2	|	c = Alta Vulnerabilidade

Matriz de Confusão – Exp. 3 - AODE				
a	b	C	←	Classificado como
300	1	5	|	a = Média Vulnerabilidade
6	0	0	|	b = Baixa Vulnerabilidade
34	0	2	|	c = Alta Vulnerabilidade

Figura 37: Matrizes de confusão do experimento 3

Fonte: Autoria própria.

6.5 EXPERIMENTO 4 - UTILIZANDO OS ATRIBUTOS DO FORMULÁRIO DE AVALIAÇÃO EM CONJUNTO COM OS ATRIBUTOS DO EXPERIMENTO 1

Esse experimento foi realizado a fim de averiguar se, utilizando as informações conjuntas do formulário de avaliação de vulnerabilidade com os atributos do conjunto de dados do experimento 1 (apresentado na Seção 6.2), seria possível obter melhores resultados.

No primeiro experimento realizado foram utilizados diversos atributos da família, sendo esses atributos: comuns de todos os membros da família, atributos do responsável familiar e atributos de todos os dependentes. Neste experimento foram adicionados novos atributos referentes ao formulário de avaliação de vulnerabilidade que são comuns a todas as famílias cadastradas. Os atributos selecionados nesse conjunto de dados foram todos os disponíveis no conjunto de dados.

Os resultados obtidos, apresentados na Tabela 11, elucidam que os algoritmos de classificação tiveram taxas de acertos muito semelhantes as do experimento 1. Porém analisando a matriz de confusão dos algoritmos constatou-se que o problema de quase sempre prever a classe majoritária ainda ocorre para esse novo conjunto de dados (Figura 38).

Tabela 11: Resultado do experimento 4 utilizando validação cruzada fator 10

Naive Bayes			AODE		
Acerto	**Erro**	**AUC**	**Acerto**	**Erro**	**AUC**
87.931%	12.069%	0.637%	87.931%	12.069%	0.628%

Fonte: Autoria própria.

Matriz de Confusão – Exp. 4 - Naive Bayes				
a	b	C	←	Classificado como
302	0	4	|	a = Média Vulnerabilidade
6	0	0	|	b = Baixa Vulnerabilidade
32	0	4	|	c = Alta Vulnerabilidade

Matriz de Confusão – Exp. 4 - AODE				
a	b	C	←	Classificado como
306	0	0	|	a = Média Vulnerabilidade
6	0	0	|	b = Baixa Vulnerabilidade
36	0	0	|	c = Alta Vulnerabilidade

Figura 38: Matrizes de confusão do experimento 4

Fonte: Autoria própria.

O presente experimento mostrou que mesmo combinando todos os dados das famílias com os dados comuns das famílias contidos no formulário de avaliação de vulnerabilidade não foi possível obter bons resultados na predição das classes.

Um item comum aos quatro experimentos realizados anteriormente é que o número de amostras por classe foi sempre o informado na Tabela 7. Ou seja, existe um grande desbalanceamento entre as classes na base de dados. Com intuito de verificar se o desbalanceamento das amostras seria o causador da dificuldade para os algoritmos de aprendizado, foram realizados dois novos experimentos com amostras balanceadas, que são apresentados nas seções 6.6 e 6.7.

Nos experimentos 5 e 6 pretendeu-se verificar se o problema dos algoritmos sempre preverem a classe majoritária também aconteceria quando a quantidade de amostras são balanceadas. Nestes experimentos, foram separadas aleatoriamente 36 amostras da classe “Alta Vulnerabilidade” e 36 amostras da classe “Média Vulnerabilidade”. A classe “Baixa Vulnerabilidade” foi descartada por possuir um número de amostras muito baixo. Após o balanceamento o conjunto de dados passou a ter 72 amostras. Assim, obteve-se com um conjunto de dados com amostras distribuídas conforme a Tabela 12.

Tabela 12: Distribuição das amostras utilizadas nos experimentos 5 e 6

Classificação	Quantidade	%
Alta vulnerabilidade	36	50
Média vulnerabilidade	36	50

Fonte: Autoria própria.

O problema a ser resolvido nos experimentos 5 e 6 foi a classificação das amostras em 2 (duas) classes “Média” e “Alta” vulnerabilidade distribuídas conforme a Tabela 12.

Nestes experimentos foram utilizados os algoritmos de classificação supervisionados (Naive Bayes e AODE) com validação cruzada fator 10.

6.6 EXPERIMENTO 5 - USANDO APENAS OS ATRIBUTOS DO FORMULÁRIO DE AVALIAÇÃO DE VULNERABILIDADE COM CLASSES BALANCEADAS

Os atributos selecionados para compor o conjunto de dados utilizados neste experimento foram aqueles comuns para todas as famílias existentes na base de dados da aplicação e ao mesmo tempo também foram utilizados no formulário de avaliação de vulnerabilidade (11 atributos). Esses atributos foram os mesmos utilizados no experimento 6.4 e podem ser observados na linha 3 da Tabela 6.

Os resultados, apresentados na Tabela 13, mostraram uma piora da taxa de acerto em relação aos experimentos anteriores. Porém ao analisar a matriz de confusão (Figura 39), pode-se perceber que os algoritmos não estavam mais classificando todas as instâncias como sendo da classe majoritária. Nesse experimento foi possível alcançar resultados razoáveis com taxa de acerto igual a 63.888% ao utilizar amostras balanceadas.

Tabela 13: Resultados do experimento 5 utilizando validação cruzada fator 10

Naive Bayes			AODE		
Acerto	**Erro**	**AUC**	**Acerto**	**Erro**	**AUC**
63.888%	36.111%	0.675%	61.111%	38.888%	0.657%

Fonte: Autoria própria.

Matriz de Confusão – Exp. 5 - Naive Bayes			
a	b	←	Classificado como
27	09	|	a = Média Vulnerabilidade
17	19	|	b = Alta Vulnerabilidade

Matriz de Confusão – Exp. 5 - AODE			
a	b	←	Classificado como
26	10	|	a = Média Vulnerabilidade
18	18	|	b = Alta Vulnerabilidade

Figura 39: Matrizes de confusão do experimento 5

Fonte: Autoria própria.

6.7 EXPERIMENTO 6 - UTILIZANDO OS ATRIBUTOS DO FORMULÁRIO DE AVALIAÇÃO DE VULNERABILIDADE EM CONJUNTO COM OS ATRIBUTOS DO EXPERIMENTO 1 COM CLASSES BALANCEADAS

O intuito desse experimento foi verificar se utilizando as informações conjuntas do formulário de avaliação de vulnerabilidade com os atributos do conjunto de dados do experimento 1 (apresentado na Seção 6.2), seria possível obter melhores resultados em comparação ao experimento 5 (apresentado na Seção 6.6) quando a quantidade de amostras fosse balanceada. Os atributos selecionados para compor esse conjunto de dados foram os mesmos utilizados no experimento 6.5 e podem ser observados na Tabela 6.

A análise dos resultados apresentados na Tabela 14 apresenta uma melhora em relação à taxa de classificação do experimento 5 (apresentado na Seção 6.6). Ao analisar a matriz de confusão (Figura 40), pôde-se perceber que os algoritmos não estavam mais classificando todas as instâncias como sendo da classe majoritária.

No presente experimento foi possível alcançar resultados razoáveis com taxa de acerto de 66.666% e AUC 0.726% ao utilizar amostras balanceadas.

Tabela 14: Resultados do experimento 6 utilizando validação cruzada fator 10

Naive Bayes			AODE		
Acerto	**Erro**	**AUC**	**Acerto**	**Erro**	**AUC**
66.666%	33.333%	0.725%	66.666%	33.333%	0.726%

Fonte: Autoria própria.

Matriz de Confusão – Exp. 6 - Naive Bayes			
a	b	←	Classificado como
27	9	|	a = Média Vulnerabilidade
15	21	|	b = Alta Vulnerabilidade

Matriz de Confusão – Exp. 6 - AODE			
a	b	←	Classificado como
27	9	|	a = Média Vulnerabilidade
15	21	|	b = Alta Vulnerabilidade

Figura 40: Matrizes de confusão do experimento 6

Fonte: Autoria própria.

6.8 ANÁLISE DOS EXPERIMENTOS PRELIMINARES

Após analisar os resultados dos experimentos pôde-se observar que os modelos utilizados sempre classificavam a amostra como membro da classe majoritária, em função do conjunto de dados de treinamento possuir amostras desbalanceadas. Após realizar um balanceamento simples das classes, retirando aleatoriamente amostras das classes, foram realizados experimentos adicionais e os resultados, apesar de terem apresentado uma taxa de acerto menor em comparação ao conjunto desbalanceado, apresentaram um resultado positivo, pois o classificador passou a predizer mais corretamente as amostras balanceadas e a medida AUC foi melhor do que para os casos desbalanceados. Diante desses fatos verificou-se a necessidade da utilização de técnicas de balanceamento no conjunto de treinamento que serão apresentadas no Capítulo 7.

Também é importante que o algoritmo utilizado para a classificação possa não apenas classificar a amostra em um dos rótulos de vulnerabilidade, mas que também seja capaz de aferir o grau de probabilidade da família estar em situação de vulnerabilidade. Uma vez que

o intuito final da solução é identificar quais famílias precisam ser prioritariamente assistidas. Desse modo, ao invés de obter as famílias rotuladas em uma das classes de vulnerabilidade, será possível obter uma estimativa de probabilidade da família ser vulnerável e consequentemente uma lista das famílias em situação de maior vulnerabilidade. Com essa estimativa será possível fornecer uma lista ordenada de prioridade de atendimento com base na probabilidade de vulnerabilidade, apoiando o processo da busca ativa. Essa lista ordenada das famílias torna-se necessária, uma vez que o município pode não possuir capacidade suficiente de atendimento para atender todas as famílias vulneráveis simultaneamente.

7 EXPERIMENTOS

Após a realização dos experimentos preliminares ficou claramente visível que o baixo número de amostras e o desbalanceamento das classes estava influenciando de maneira negativa os resultados. A predição das classes estava sendo fortemente influenciada pela classe majoritária enquanto a classe de interesse, a classe minoritária, estava sendo predita de forma incorreta.

A fim de obter uma melhor classificação da classe de interesse foram realizados novos experimentos com métodos de balanceamento de amostras entre as classes propostos na literatura e citados na Seção 4, ROS (BATISTA et al., 2004), SMOTE (CHAWLA et al., 2002), SMOTE+Tomek Links (BATISTA et al., 2004), SMOTE+ENN (BATISTA et al., 2004), Bordeline-SMOTE (HAN et al., 2005), Safe-Level-SMOTE (BUNKHUMPORNPAT et al., 2009), SPIDER (STEFANOWSKI; WILK, 2008), SPIDER2 (NAPIERALA et al., 2010).

O principal propósito deste novo conjunto de experimentos foi verificar se a predição após realizado o balanceamento das amostras entre as classes no conjunto de dados de treinamento poderia ser melhor do que os resultados obtidos nos experimentos preliminares realizados com os conjuntos de dados desbalanceados. Também pretendeu-se verificar se seria possível realizar corretamente a predição das amostras da classe minoritária que é a classe de interesse representando as famílias em situação de alta vulnerabilidade social.

Nessa seção serão apresentados os detalhes das bases de dados utilizadas nestes experimentos, o protocolo experimental e as métricas de avaliação utilizadas para avaliar os resultados obtidos.

7.1 CONFIGURAÇÃO EXPERIMENTAL

As informações utilizadas nestes experimentos foram extraídas do sistema IRSAS em dois momentos. As amostras obtidas destas bases de dados são categorizadas em três diferentes classes (Baixa, Média e Alta) de acordo com a classificação de vulnerabilidade das famílias

como mostrada na Tabela 1.

A primeira base de dados, referida a frente neste trabalho com Database 01 foi coletada em Março de 2013 e possui 348 amostras distribuídas em três classes: Baixa (6 amostras), Média (306 amostras) e Alta (36 amostras). Esta foi a base de dados utilizada nos experimentos preliminares descritos na Seção 6.

A segunda base de dados, referida a frente neste trabalho como Database 02 foi coletada em Julho de 2014 e possui 1.504 amostras distribuídas em três classes: Baixa (137 amostras), Média (1.081 amostras) e Alta (286 amostras).

A distribuição das classes nas duas bases de dados é apresentada na Tabela 15.

Tabela 15: Distribuição das amostras nas duas bases de dados de acordo com a classe de vulnerabilidade

Base de dados	Baixa	Média	Alta
Database 01	6	306	36
Database 02	137	1.081	286

Fonte: Autoria própria.

A segunda base de dados contém as amostras iniciais da primeira base de dados mais as novas amostras coletadas pelo sistema durante o período de tempo passado entre uma coleta e outra.

A classe de interesse é a Alta Vulnerabilidade pois representa as famílias em maior vulnerabilidade e risco social e por isso precisam ser assistidas prioritariamente. Deste modo, como a Database 02 contém um maior número de amostras da classe Alta Vulnerabilidade, é esperado que o resultado da predição dessas amostras seja melhor do que o obtido nos experimentos realizados com a Database 01.

Para realização dos experimentos balanceados optou-se por utilizar o conjunto de dados Dataset_04 conforme Tabela 6 que possui todos os atributos disponíveis na base de dados, ambas as bases de dados contém exatamente os mesmos atributos que são listados na Tabela 5 onde cada atributo é identificado com um código numérico disposto na primeira coluna da tabela.

7.2 PROTOCOLO EXPERIMENTAL

Em função do baixo número de amostras optou-se por utilizar nos experimentos a validação cruzada fator 5 que é o procedimento comumente utilizado nos trabalhos da área

((RAMENTOL et al., 2012), (LUENGO et al., 2011), (LÓPEZ et al., 2013), (LÓPEZ et al., 2014), (MAHESHWARI et al., 2011), (LÓPEZ et al., 2012), (GALAR et al., 2013) e (GALAR et al., 2012)).

Alguns métodos de balanceamento foram utilizados para verificar se melhores resultados poderiam ser obtidos e para isso foi necessário realizar o particionamento dos conjuntos de dados em 5 partes antes de realizar a aplicação dos métodos de balanceamento. A Figura 41 ilustra como o processo descrito a seguir foi realizado. Primeiro o conjunto de dados foi particionado em 5 partições. Cada partição foi dividida em conjunto de treinamento e teste. O conjunto de treinamento foi então balanceado para que a distribuição das amostras entre as classes fossem iguais. Nada foi feito com o conjunto de teste. Após estes passos o conjunto balanceado de treinamento foi utilizado para gerar um modelo através de um algoritmo de classificação e então este modelo foi testado com base no conjunto de teste. A média dos resultados das 5 partições foi calculada e então o resultado final foi obtido.

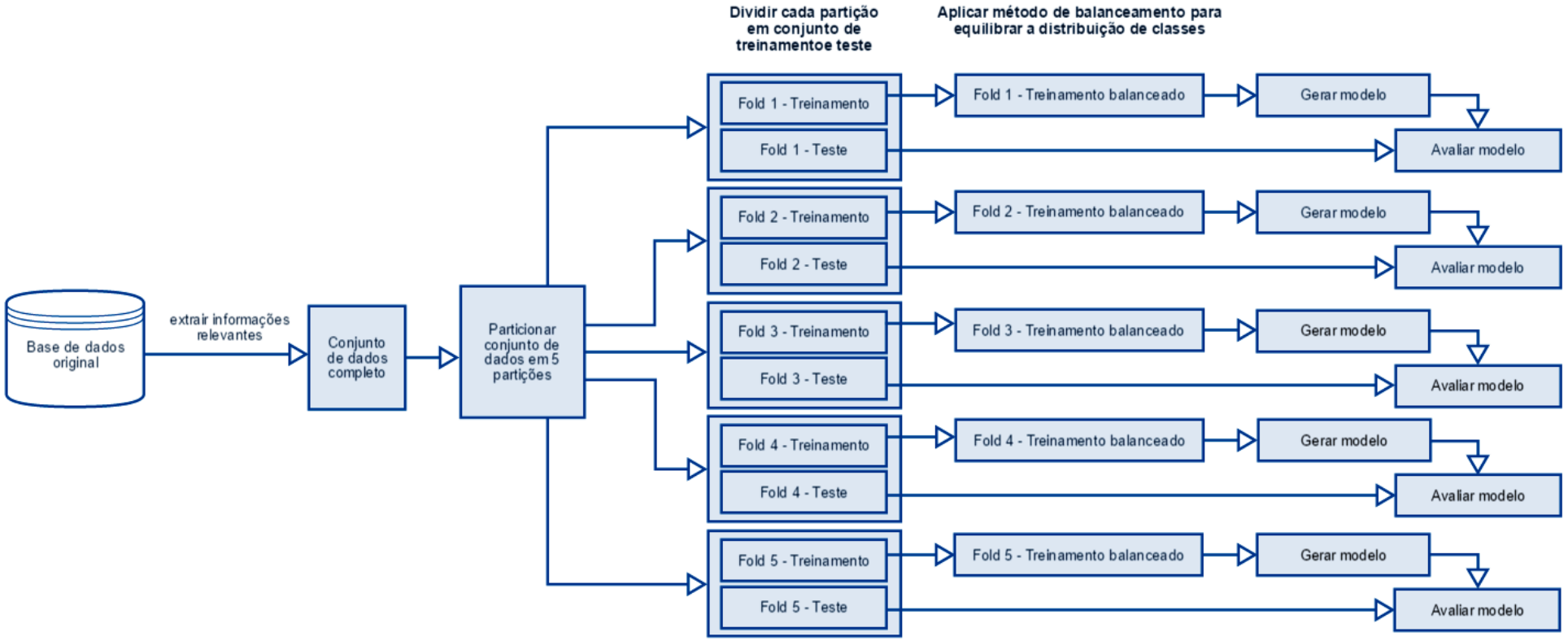


Figura 41: Protocolo experimental

Fonte: Autoria própria.

Como alguns métodos de balanceamento funcionam apenas com classes binárias e a maioria dos trabalhos relacionados também utilizam problemas de apenas duas classes, decidiu-se por realizar duas abordagens diferentes para transformar os conjuntos de dados em conjuntos binários.

A primeira abordagem foi manter a classe de interesse (Alta Vulnerabilidade) e juntar as classes (Baixa e Media Vulnerabilidade) em uma única classe. Este procedimento é comum na área ((CHAWLA et al., 2002), (BATISTA et al., 2004), (STEFANOWSKI; WILK, 2008), (BUNKHUMPORNPAT et al., 2009), (RAMENTOL et al., 2012) e (GALAR et al., 2012)). Para representar essa abordagem o prefixo ***M***_ foi colocado no nome do conjunto de dados representando que as classes restantes foram combinadas (*merged*).

A segunda abordagem foi manter a classe de interesse (Alta Vulnerabilidade) e a classe majoritária (Média Vulnerabilidade), descartando a classe de Baixa Vulnerabilidade restante, como feito em Chawla et al. (2002). Para representar essa abordagem o prefixo ***D***_ foi colocado no nome do conjunto de dados representando que as classes restantes foram descartadas.

Essas abordagens resultaram em conjuntos de dados com apenas duas classes para cada conjunto de dados.

Por exemplo, o Dataset_04 que originalmente possuía 3 classes, após a aplicação das abordagens descritas resulta em dois conjuntos de dados binários:

M_Dataset_04 - O prefixo **M**_ significa que a classe Baixa Vulnerabilidade foi combinada com a classe Média Vulnerabilidade e se tornou uma única classe chamada Baixa/Média Vulnerabilidade.

D_Dataset_04 - O prefixo **D**_ significa que a classe Baixa Vulnerabilidade foi descartada.

A implementação dos métodos de balanceamento utilizados neste trabalho foram tomadas da ferramenta de código aberto KEEL apresentada na Seção 4.9. Os parâmetros de configuração utilizados em cada um dos métodos são apresentados na Tabela 16. Os métodos de balanceamento utilizados foram os seguintes:

- **ROS** (Random over-sampling) (BATISTA et al., 2004).
- **SMOTE** (Synthetic Minority Over-sampling TEchnique) (CHAWLA et al., 2002).
- **SMOTE+Tomek Links** (Synthetic Minority Over-sampling TEchnique + Tomek's modification of Condensed Nearest Neighbor) (BATISTA et al., 2004).

- **SMOTE+ENN** (Synthetic Minority Over-sampling TEchnique + Edited Nearest Neighbor) (BATISTA et al., 2004).
- **Borderline-SMOTE** (Borderline-Synthetic Minority Over-sampling TEchnique) (HAN et al., 2005).
- **Safe-Level-SMOTE** (Safe Level Synthetic Minority Over-sampling TEchnique) (BUNKHUMPORNPAT et al., 2009).
- **SPIDER** (Selective Preprocessing of Imbalanced Data) (STEFANOWSKI; WILK, 2008).
- **SPIDER2** (Selective Preprocessing of Imbalanced Data 2) (NAPIERALA et al., 2010).

Após realizar o balanceamento dos conjuntos de dados de treinamento com a ferramenta KEEL foi utilizada a ferramenta WEKA apresentada na Seção 3.8 para discretizar os atributos numéricos através do filtro *weka.filters.supervised.attribute.Discretize* e posteriormente para realizar a etapa de classificação utilizando os seguintes algoritmos:

- **NB** (Naive Bayes) (DOUGHERTY, 2013).
- **AODE** (Averaged One-Dependence Estimators) (WEBB et al., 2005).

Os resultados dos experimentos podem ser observados nas Tabelas 17, 18, 19 e 20.

Tabela 16: Parâmetros de configuração dos métodos de balanceamento na ferramenta KEEL

Parâmetro	**SMOTE**	**SMOTE+ Tomek Links**	**SMOTE+ENN**	**Borderline-SMOTE**	**Safe-Level-SMOTE**	**SPIDER**	**SPIDER2**
Number of Neighbors	5	5	5	5	5	3	3
Distance Function	HVDM	HVDM	HVDM	HVDM	HVDM	HVDM	HVDM
Type of SMOTE	both	both	both	both	both	–	–
Balancing	YES	YES	YES	YES	YES	–	–
Quantity of generated examples	1	1	1	1	1	–	–
Type of Interpolation	standard	–	–	standard	standard	–	–
Alpha	0.5	–	–	0.5	0.5	–	–
Mu	0.5	–	–	0.5	0.5	–	–
Number of Neighbors ENN	–	–	3	–	–	–	–
Number of Neighbors for considering a instance BORDER	–	–	–	3	–	–	–
Type of Borderline SMOTE	–	–	–	1	–	–	–
Preprocessing Option	–	–	–	–	–	WEAK	–
relabel	–	–	–	–	–	–	true
ampl	–	–	–	–	–	–	strong

Fonte: Autoria própria.

Tabela 17: Resultados Dataset M_04 Database 01

	Naive Bayes				AODE			
Método	**Precisão**	**Recall**	**F-measure**	**DCG**	**Precisão**	**Recall**	**F-measure**	**DCG**
Original	0,28 ±0,20	0,33 ±0,12	0,29 ±0,14	1,52 ±0,75	0,20 ±0,45	0,03 ±0,06	0,05 ±0,11	0,20 ±0,45
ROS	0,21 ±0,09	0,46 ±0,26	0,29 ±0,13	1,52 ±0,90	0,12 ±0,15	0,16 ±0,20	0,14 ±0,17	0,67 ±0,83
SMOTE	0,20 ±0,09	0,58 ±0,23	0,30 ±0,12	1,60 ±0,93	0,12 ±0,13	0,27 ±0,31	0,17 ±0,19	0,98 ±1,09
SMOTE_TL	0,21 ±0,08	0,60 ±0,24	0,31 ±0,12	1,80 ±0,94	0,13 ±0,14	0,30 ±0,35	0,18 ±0,20	1,06 ±1,24
SMOTE_ENN	0,20 ±0,30	0,08 ±0,11	0,11 ±0,16	0,53 ±0,75	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00
Borderline_SMOTE	0,24 ±0,17	0,27 ±0,20	0,25 ±0,19	1,23 ±0,78	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00
Safe_Level_SMOTE	0,18 ±0,08	0,52 ±0,25	0,27 ±0,12	1,39 ±0,86	0,09 ±0,06	0,28 ±0,22	0,14 ±0,10	0,85 ±0,85
SPIDER	0,23 ±0,13	0,52 ±0,27	0,31 ±0,17	1,99 ±1,05	0,22 ±0,22	0,13 ±0,15	0,16 ±0,18	0,81 ±0,86
SPIDER2	0,23 ±0,14	0,55 ±0,28	0,32 ±0,17	1,88 ±0,95	0,20 ±0,16	0,21 ±0,19	0,21 ±0,17	1,13 ±1,00

Fonte: Autoria própria.

Tabela 18: Resultados Dataset D_04 Database 01

	Naive Bayes				AODE			
Método	**Precisão**	**Recall**	**F-measure**	**DCG**	**Precisão**	**Recall**	**F-measure**	**DCG**
Original	0,22 ±0,13	0,36 ±0,26	0,27 ±0,16	1,52 ±0,95	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00
ROS	0,25 ±0,09	0,50 ±0,07	0,33 ±0,09	1,61 ±0,53	0,11 ±0,11	0,11 ±0,12	0,11 ±0,10	0,73 ±0,83
SMOTE	0,19 ±0,04	0,53 ±0,13	0,28 ±0,06	1,65 ±0,62	0,24 ±0,06	0,36 ±0,16	0,28 ±0,07	1,27 ±0,24
SMOTE_TL	0,21 ±0,05	0,59 ±0,16	0,30 ±0,08	1,67 ±0,66	0,21 ±0,16	0,30 ±0,21	0,24 ±0,16	1,11 ±0,67
SMOTE_ENN	0,40 ±0,42	0,08 ±0,08	0,13 ±0,12	0,60 ±0,55	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00
Borderline_SMOTE	0,25 ±0,03	0,28 ±0,11	0,25 ±0,05	1,13 ±0,34	0,20 ±0,45	0,03 ±0,06	0,04 ±0,10	0,20 ±0,45
Safe_Level_SMOTE	0,19 ±0,04	0,53 ±0,13	0,28 ±0,05	1,40 ±0,51	0,18 ±0,10	0,31 ±0,12	0,22 ±0,09	0,81 ±0,24
SPIDER	0,22 ±0,04	0,56 ±0,11	0,31 ±0,05	1,87 ±0,44	0,33 ±0,20	0,20 ±0,16	0,22 ±0,15	1,06 ±0,68
SPIDER2	0,22 ±0,05	0,53 ±0,06	0,31 ±0,06	1,91 ±0,39	0,29 ±0,21	0,17 ±0,16	0,18 ±0,11	0,95 ±0,62

Fonte: Autoria própria.

Tabela 19: Resultados Dataset M_04 Database 02

	Naive Bayes				AODE			
Método	**Precisão**	**Recall**	**F-measure**	**DCG**	**Precisão**	**Recall**	**F-measure**	**DCG**
Original	0,33 ±0,06	0,49 ±0,08	0,39 ±0,07	6,94 ±1,12	0,57 ±0,11	0,16 ±0,07	0,24 ±0,08	4,25 ±1,11
ROS	0,31 ±0,05	0,52 ±0,09	0,38 ±0,06	7,30 ±1,16	0,36 ±0,04	0,48 ±0,03	0,41 ±0,03	7,69 ±0,91
SMOTE	0,30 ±0,05	0,54 ±0,11	0,38 ±0,07	7,40 ±1,32	0,32 ±0,03	0,45 ±0,06	0,38 ±0,04	7,31 ±0,65
SMOTE_TL	0,21 ±0,06	0,91 ±0,16	0,34 ±0,05	8,81 ±0,89	0,22 ±0,06	0,90 ±0,22	0,33 ±0,04	8,83 ±1,30
SMOTE_ENN	0,72 ±0,22	0,04 ±0,02	0,08 ±0,04	1,79 ±0,59	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00
Borderline_SMOTE	0,31 ±0,07	0,49 ±0,11	0,38 ±0,08	6,98 ±1,51	0,35 ±0,05	0,36 ±0,07	0,35 ±0,04	6,35 ±0,71
Safe_Level_SMOTE	0,28 ±0,05	0,57 ±0,11	0,38 ±0,07	7,46 ±1,69	0,31 ±0,04	0,58 ±0,05	0,40 ±0,04	8,44 ±1,07
SPIDER	0,29 ±0,05	0,55 ±0,11	0,38 ±0,07	7,53 ±1,36	0,43 ±0,06	0,41 ±0,06	0,42 ±0,05	6,80 ±1,11
SPIDER2	0,29 ±0,05	0,58 ±0,11	0,39 ±0,07	7,78 ±1,69	0,36 ±0,03	0,51 ±0,04	0,42 ±0,03	3,61 ±3,94

Fonte: Autoria própria.

Tabela 20: Resultados Dataset D_04 Database 02

	Naive Bayes				AODE			
Método	**Precisão**	**Recall**	**F-measure**	**DCG**	**Precisão**	**Recall**	**F-measure**	**DCG**
Original	0,22 ±0,13	0,36 ±0,26	0,27 ±0,16	1,52 ±0,95	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00
ROS	0,25 ±0,09	0,50 ±0,07	0,33 ±0,09	1,61 ±0,53	0,11 ±0,11	0,11 ±0,12	0,11 ±0,10	0,73 ±0,83
SMOTE	0,19 ±0,04	0,53 ±0,13	0,28 ±0,06	1,65 ±0,62	0,24 ±0,06	0,36 ±0,16	0,28 ±0,07	1,27 ±0,24
SMOTE_TL	0,21 ±0,05	0,59 ±0,16	0,30 ±0,08	1,67 ±0,66	0,21 ±0,16	0,30 ±0,21	0,24 ±0,16	1,11 ±0,67
SMOTE_ENN	0,40 ±0,42	0,08 ±0,08	0,13 ±0,12	0,60 ±0,55	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00	0,00 ±0,00
Borderline_SMOTE	0,25 ±0,03	0,28 ±0,11	0,25 ±0,05	1,13 ±0,34	0,20 ±0,45	0,03 ±0,06	0,04 ±0,10	0,20 ±0,45
Safe_Level_SMOTE	0,19 ±0,04	0,53 ±0,13	0,28 ±0,05	1,40 ±0,51	0,18 ±0,10	0,31 ±0,12	0,22 ±0,09	0,81 ±0,24
SPIDER	0,22 ±0,04	0,56 ±0,11	0,31 ±0,05	1,87 ±0,44	0,33 ±0,20	0,20 ±0,16	0,22 ±0,15	1,06 ±0,68
SPIDER2	0,22 ±0,05	0,53 ±0,06	0,31 ±0,06	1,91 ±0,39	0,29 ±0,21	0,17 ±0,16	0,18 ±0,11	0,95 ±0,62

Fonte: Autoria própria.

7.3 AVALIAÇÃO DOS EXPERIMENTOS

Nesta seção serão apresentadas as avaliações dos experimentos realizados com os métodos de balanceamento das amostras nos conjuntos de dados. Inicialmente foram utilizadas as medidas padrões da área (Precisão, *Recall*, e F-measure) para análise de desempenho com conjuntos de dados desbalanceados. As avaliações através dessas medidas não foram suficientes para identificar qual método de balanceamento obteve o melhor resultado. Em função disso, uma nova forma de avaliação utilizando o *Recall* e a medida DCG foi proposta para avaliar os resultados. Por fim, é feita uma análise geral dos resultados obtidos nos experimentos.

7.3.1 AVALIAÇÃO ATRAVÉS DA PRECISÃO

A métrica de avaliação Precisão, apresentada na Seção 3.6.1, é comumente utilizada na avaliação de resultados de classificadores pois permite identificar a quantidade de amostras positivas classificadas corretamente sobre o total de amostras classificadas como positivas. Dessa maneira, no cenário desse trabalho, é possível avaliar a quantidade de famílias que foram classificadas corretamente como Alta Vulnerabilidade sobre o número total de famílias classificadas como Alta Vulnerabilidade, em outras palavras, a medida Precisão da classe de Alta Vulnerabilidade indica dentre todas as famílias classificadas como Alta Vulnerabilidade aquelas que realmente pertencem a esta classe.

Ao avaliar os resultados dos experimentos com essa medida foi observado que a métrica da Precisão, quando utilizada de forma isolada, não é suficiente para apontar qual método obteve o melhor resultado entre todos os utilizados. Por exemplo, no caso do classificador Naive Bayes quando utilizado com o método de balanceamento SMOTE_ENN no conjunto de dados M_04 da Database 02, onde os resultados estão apresentados na Tabela 21. Esta configuração obteve o melhor resultado geral de precisão para esse conjunto de dados atingindo o valor de 72%, porém ao verificar os valores das medidas *Recall* e F-measure pode-se observar que os valores alcançados por essa configuração não foram tão bons quanto a Precisão, obtendo valores de 4% e 8% respectivamente.

Tabela 21: Resultado do classificador Naive Bayes para o Dataset M_04 da Database 02 com o método de balanceamento SMOTE_ENN

	Naive Bayes		
Método	**Precisão**	**Recall**	**F-measure**
SMOTE_ENN	**0,72 ±0,22**	0,04 ±0,02	0,08 ±0,04

Fonte: Autoria própria.

A Figura 42 apresenta a média das matrizes de confusão dos resultados das 5 partições utilizadas na validação cruzada do experimento dado com exemplo.

Média das matrizes de confusão

a	b	←	Classificado como
242,4	1,2	|	a = Baixa Média Vulnerabilidade
54,8	2,4	|	b = Alta Vulnerabilidade

Figura 42: Média das matrizes de confusão dos resultados do classificador Naive Bayes com o conjunto de dados M_04 da Database 02 com método de balanceamento SMOTE_ENN

Fonte: Autoria própria.

Ao verificar os resultados apresentados na Figura 42 fica evidente que mesmo obtendo uma taxa de Precisão de 72% para a classe Alta Vulnerabilidade o classificador rotulou quase que a totalidade das amostras da classe de Alta Vulnerabilidade como pertencentes a classe de Baixa/Média Vulnerabilidade. 54,8 amostras da classe Alta Vulnerabilidade foram classificadas como Baixa/Média Vulnerabilidade em um total de 57,2 amostras.

Na prática o comportamento dessa configuração possui um ponto positivo, pois não está identificando famílias de Baixa e Média Vulnerabilidade como pertencentes a classe Alta Vulnerabilidade, apenas 1,2 amostras da classe Baixa/Média Vulnerabilidade foram classificadas incorretamente como Alta Vulnerabilidade em um total de 243,6 amostras. Porém, as famílias de interesse que estão em alta vulnerabilidade social não estão sendo classificadas corretamente o que causa uma situação de invisibilidade dessas famílias e faz com os resultados obtidos por esse experimento sejam ruins.

7.3.2 AVALIAÇÃO ATRAVÉS DO *RECALL*

A métrica *Recall*, apresentada na Seção 3.6.1, também conhecida por TPR (*True Positive Rate*) ou Sensitividade é outra medida utilizada para avaliar os resultados dos classificadores no aprendizado de máquina. Esta medida avalia a quantidade de amostras positivas classificadas corretamente sobre o total de amostras positivas. Em outras palavras essa medida avalia a quantidade de famílias classificadas como Alta Vulnerabilidade do total de famílias que pertencem a classe Alta Vulnerabilidade.

Ao avaliar os resultados dos experimentos com essa medida foi percebido que o *Recall*,

quando utilizado de forma isolada, também não é suficiente para apontar qual método obteve o melhor resultado entre todos os utilizados. Por exemplo, no caso do classificador Naive Bayes utilizado com o método de balanceamento SMOTE_ENN no conjunto de dados D_04 da Database 02, (onde os resultados estão apresentados na Tabela 22), obteve o melhor resultado geral de *Recall* para esse conjunto de dados atingindo o valor de 98%. Porém ao verificar os valores das medidas de Precisão e F-measure pode-se observar que os valores alcançados por essa configuração não foram tão bons quanto o *Recall*, obtendo valores de 21% e 34% respectivamente.

Tabela 22: Resultado do classificador Naive Bayes para o Dataset D_04 da Database 02 com o método de balanceamento SMOTE_ENN

	Naive Bayes		
Método	**Precisão**	**Recall**	**F-measure**
SMOTE_ENN	0,21 ±0,00	**0,98 ±0,01**	0,34 ±0,00

Fonte: Autoria própria.

A Figura 43 apresenta a média das matrizes de confusão dos resultados das 5 partições utilizadas na validação cruzada do experimento dado com exemplo.

Média das matrizes de confusão

a	b	←	Classificado como
0,8	215	|	a = Baixa Média Vulnerabilidade
1	56,2	|	b = Alta Vulnerabilidade

Figura 43: Média das matrizes de confusão dos resultados do classificador Naive Bayes com o conjunto de dados D_04 da Database 02 com método de balanceamento SMOTE_ENN

Fonte: Autoria própria.

Quando analisados os resultados apresentados na matriz de confusão pode-se observar que a taxa de *Recall* de 98% da classe Alta Vulnerabilidade foi atingida devido ao fato do classificador rotular quase sempre as amostras como pertencentes a esta classe. Apenas 0,8 amostras da classe Baixa/Média Vulnerabilidade foram classificadas corretamente de um total de 215,8 amostras desta classe. Já a classe Alta Vulnerabilidade teve apenas 1 amostra classificada incorretamente de um total de 57,2 amostras desta classe.

Na prática o comportamento dessa configuração não é bom pois uma grande quantidade de famílias que não estão em situação de alta vulnerabilidade social são classificadas como Alta Vulnerabilidade. Apesar de todas as famílias de alta vulnerabilidade social terem

sido classificadas corretamente, o fato das demais famílias também terem sido classificadas como pertencentes a classe Alta Vulnerabilidade inviabiliza a utilização dessa configuração pois resultaria em um trabalho excessivo e desnecessário para os técnicos da assistência social verificarem essas famílias.

7.3.3 AVALIAÇÃO ATRAVÉS DO F-MEASURE

Ao analisar os resultados dos experimentos realizados utilizando a medida F-measure da classe Alta vulnerabilidade, pode-se observar que nenhum dos experimentos realizados obteve um bom resultado quando avaliado por essa medida. Esses resultados podem ser observados nas Tabelas 17, 18, 19 e 20. O melhor resultado obtido para a medida F-measure da classe Alta Vulnerabilidade nesta configuração foi de 42% pelo classificador AODE com o conjunto de dados M_04 da Database 02 com o método de balanceamento SPIDER. Os resultados do experimento citado podem ser observados na Tabela 23.

Tabela 23: Resultado do classificador AODE para o Dataset M_04 da Database 02 com o método de balanceamento SPIDER

	AODE		
Método	**Precisão**	**Recall**	**F-measure**
SPIDER	0,43 ±0,06	0,41 ±0,06	**0,42 ±0,05**

Fonte: Autoria própria.

A Figura 44 apresenta a média das matrizes de confusão dos resultados das 5 partições utilizadas na validação cruzada do experimento dado como exemplo.

Média das matrizes de confusão

a	b	←	Classificado como
192,8	50,8	|	a = Baixa Média Vulnerabilidade
28,2	29	|	b = Alta Vulnerabilidade

Figura 44: Média das matrizes de confusão dos resultados do classificador AODE com o conjunto de dados M_04 da Database 02 com método de balanceamento SPIDER

Fonte: Autoria própria.

Quando analisados os resultados apresentados na Figura 44 pode-se observar que o classificador não previu todas as amostras como sendo de uma única classe mas também não

foi capaz de prever a maioria das amostras da classe de interesse minoritária. Para a classe Baixa/Média Vulnerabilidade foram classificadas corretamente, na média 192,8 amostras de um total de 243,6 amostras desta classe. Para a classe Alta Vulnerabilidade foram classificadas corretamente 29 amostras de um total de 57,2 amostras desta classe.

Na prática o comportamento dessa configuração não é desejável pois apenas uma parte das famílias em situação de alta vulnerabilidade social serão identificadas pelo classificador, o que faria com que várias famílias necessitadas de assistência permanecessem invisíveis aos técnicos da assistência social.

7.3.4 PROPOSTA DE UMA NOVA FORMA DE AVALIAÇÃO

Em face das dificuldades encontradas para analisar os resultados dos experimentos e encontrar a melhor configuração capaz de identificar as famílias em situação de alta vulnerabilidade social, buscou-se outras formas para proceder a avaliação dos resultados.

Ao utilizar algoritmos de classificação probabilísticos, é possível obter uma lista das amostras de um conjunto de dados com o valor da probabilidade daquela amostra pertencer a uma classe específica. Neste caso, esta lista foi gerada e ordenada pelo valor da probabilidade da amostra pertencer à classe Alta Vulnerabilidade. As amostras com as maiores probabilidades de pertencerem a classe Alta Vulnerabilidade estão no topo da lista e as com menores probabilidades estão no final. Esta lista ordenada permitiu utilizar uma medida de qualidade de *ranking* para comparar os resultados obtidos.

Para comparar a qualidade de *ranking* utilizou-se a medida DCG (*Discounted Cumulative Gain*) proposta por JÄrvelin e KekÄlÄinen (2002) e apresentada na Seção 3.7.

Ao aplicar o DCG no problema real tratado neste trabalho, é muito importante que a lista ordenada tenha as famílias da classe Alta Vulnerabilidade no topo, pois estas famílias necessitam ser assistidas prioritariamente. Em um cenário onde o município tenha mais famílias necessitando de assistência do que a sua capacidade atual de atendimento, esta lista ordenada pode ser utilizada para organizar as prioridades de atendimento.

Para compreender como o DCG pode ser utilizado para comparar os resultados dos experimentos realizados, o seguinte exemplo é apresentado:

Em uma lista de 6 (seis) famílias classificadas como pertencentes à classe Alta Vulnerabilidade por um algoritmo qualquer.

$F_1, F_2, F_3, F_4, F_5, F_6$

Baseado no atributo classe de cada amostra é configurada a relevância do item para 1 (um) quando a família realmente for da classe Alta Vulnerabilidade e 0 (zero) para quando a família não for desta classe.

Após configurar a relevância de cada item o resultado é:

$1, 0, 1, 1, 0, 0$

Isto significa que a família 1 possui a relevância 1 porque é realmente da classe Alta Vulnerabilidade, família 2 possui a relevância 0 porque não é da classe Alta Vulnerabilidade, e assim por diante.

Utilizando a escala logarítmica para redução, o DCG para cada item do resultado ordenado é apresentado na última coluna da Tabela 24.

Tabela 24: Exemplo DCG

i	Classe real	Classe predita	rel_i	$\log_2 i$	$\frac{rel_i}{\log_2 i}$
1	Alta Vulnerabilidade	Alta Vulnerabilidade	1	0	N/A
2	Baixa Vulnerabilidade	Alta Vulnerabilidade	0	1	0
3	Alta Vulnerabilidade	Alta Vulnerabilidade	1	1,585	0,63
4	Alta Vulnerabilidade	Alta Vulnerabilidade	1	2,0	0,5
5	Baixa Vulnerabilidade	Alta Vulnerabilidade	0	2,322	0
6	Média Vulnerabilidade	Alta Vulnerabilidade	0	2,584	0

Fonte: Autoria própria.

Dessa maneira, o DCG_6 para essa lista ordenada é:

$$DCG_6 = rel_1 + \sum_{i=2}^{6} \frac{rel_i}{log_2(i)} = 1 + (0 + 0,63 + 0,5 + 0 + 0) = 2,13 \tag{22}$$

Se fosse realizada uma troca de ordem das famílias F_2 e F_3, isto resultaria em um valor maior do DCG, porque uma família mais relevante seria posicionada primeiramente no *ranking*, isto é, uma família mais relevante não sofre mais o desconto da posição 3 por estar posicionada em um ponto mais alto do *ranking*. O valor do DCG de cada item do novo resultado ordenado é apresentado na última coluna da Tabela 25. Desse modo o valor final acumulado do DCG seria de 2,5 ao invés de 2,13 conforme apresentado na Equação 23 .

$$DCG_6 = rel_1 + \sum_{i=2}^{6} \frac{rel_i}{log_2(i)} = 1 + (1 + 0 + 0,5 + 0 + 0) = 2,5 \tag{23}$$

Tabela 25: Exemplo DCG 2

i	Classe real	Classe predita	rel_i	$\log_2 i$	$\frac{rel_i}{\log_2 i}$
1	Alta Vulnerabilidade	Alta Vulnerabilidade	1	0	N/A
2	Alta Vulnerabilidade	Alta Vulnerabilidade	1	1	1
3	Baixa Vulnerabilidade	Alta Vulnerabilidade	0	1,585	0
4	Alta Vulnerabilidade	Alta Vulnerabilidade	1	2,0	0,5
5	Baixa Vulnerabilidade	Alta Vulnerabilidade	0	2,322	0
6	Média Vulnerabilidade	Alta Vulnerabilidade	0	2,584	0

Fonte: Autoria própria.

A medida DCG torna possível comparar os resultados dos métodos de balanceamento e classificação utilizados nos experimentos deste trabalho a fim de verificar qual deles produzem os melhores resultado neste cenário. Dessa maneira, quanto maior o valor do DCG para o método, mais ele estará acertando a probabilidade das amostras da classe Alta vulnerabilidade serem preditas como Alta Vulnerabilidade. Os resultados do DCG de cada experimento são apresentados nas Tabelas 17, 18, 19 e 20.

7.3.5 AVALIAÇÃO ATRAVÉS DO DCG

Através do DCG pôde-se comparar os resultados de duas listas ordenadas de resultados geradas pelos classificadores. Ao avaliar os resultados dos experimentos com essa medida foi percebido que o DCG quando utilizado de forma isolada, também não é suficiente para identificar qual método obteve o melhor resultado entre todos os utilizados. Conforme apresentado na Tabela 26, pode-se observar que os resultados da medida DCG para os métodos de balanceamento Safe_Level_SMOTE e SMOTE_TL com o classificador AODE foram muito parecidos onde o primeiro obteve valor de 8,44 e o segundo 8,83. Porém ao verificar o valor do *Recall* nota-se que o método de balanceamento SMOTE_TL foi superior pois obteve 90% de *Recall* enquanto o método Safe_Level_SMOTE obteve 58%.

Tabela 26: Resultados parciais do classificador AODE para o Dataset M_04 da Database 02

	AODE			
Método	**Precision**	**Recall**	**F-measure**	**DCG**
Safe_Level_SMOTE	0,31 ±0,04	0,58 ±0,05	0,40 ±0,04	8,44 ±1,07
SMOTE_TL	0,22 ±0,06	0,90 ±0,22	0,33 ±0,04	8,83 ±1,30

Fonte: Autoria própria.

Através dessa análise é possível perceber que um método pode obter um valor DCG alto mesmo sem prever a maioria das amostras da classe Alta Vulnerabilidade desde que as amostras corretamente classificadas desta classe estejam no topo da lista ordenada em função da sua probabilidade de pertencer à classe Alta Vulnerabilidade.

Para encontrar as famílias em situação de alta vulnerabilidade social é fundamental que o método escolhido seja capaz de maximizar o *Recall* da classe Alta Vulnerabilidade pois as famílias nessa situação precisam ser identificadas. Por esse motivo a análise dos métodos através da medida DCG de forma isolada se mostrou ineficiente.

7.3.6 UMA NOVA FORMA DE AVALIAÇÃO ATRAVÉS DO *RECALL* E DCG

Uma vez que as medidas padrões para avaliação dos resultados não se mostraram suficientes para identificar os melhores métodos para resolução do problema abordado neste trabalho, foram buscadas outras formas para avaliar quais métodos são capazes de maximizar a classificação correta das amostras como sendo da classe de Alta Vulnerabilidade.

Porém, levando em consideração o problema prático de que além de classificar corretamente as famílias em uma das classes de vulnerabilidade é importante que o método consiga identificar quais famílias estão mais propensas a estar em alto grau de alta vulnerabilidade, passou-se também a ser analisado os resultados dos *rankings*. A partir disso foi gerada uma forma combinada de avaliação dos valores gerados por ambas as medidas do *ranking* e *Recall* da classe de interesse. Como alternativa para avaliar os resultados dos experimentos foram utilizadas duas medidas em conjunto, o *Recall* da classe Alta Vulnerabilidade e a medida DCG (*Discounted Cumulative Gain*).

A medida *Recall* foi escolhida pois ela maximiza a quantidade de amostras da classe Alta Vulnerabilidade classificadas corretamente pelo algoritmo. Na prática, o custo de classificar uma amostra das classes Baixa Vulnerabilidade e Media Vulnerabilidade como sendo da classe Alta Vulnerabilidade é menor do que não identificar que uma família pertence a classe Alta Vulnerabilidade. Ao maximizar o valor do *Recall* da classe Alta Vulnerabilidade os algoritmos experimentados passaram a baixar significativamente a Precisão, ou seja, estão classificando corretamente as amostras da classe Alta Vulnerabilidade mas também estão classificando várias amostras das outras classes como sendo Alta Vulnerabilidade. Para contornar esse problema foi proposto a utilização da medida DCG em conjunto com o *Recall* de forma a tornar possível identificar quais algoritmos estão produzindo resultados onde as famílias que realmente estão em situação de Alta Vulnerabilidade apareçam primeiro na lista ordenada pela

probabilidade enquanto as famílias de outas classes que foram preditas incorretamente como sendo da classe Alta Vulnerabilidade apareçam no final da lista ordenada.

Ao utilizar a medida DCG foi possível identificar que algoritmos com a mesma taxa de *Recall* e Precisão produziam resultados diferentes quando avaliados pelo DCG, ou seja, apesar de possuírem taxas de *Recall* e Precisão semelhantes alguns métodos são capazes de prever com maior probabilidade as amostras que realmente são da classe Alta Vulnerabilidade. Por exemplo, nos casos apresentados na Tabela 27, onde os métodos de balanceamento **ROS** e **SPIDER2** possuem praticamente as mesmas taxas de Precisão e *Recall* porém no caso do **SPIDER2** a medida DCG foi de 3,82 enquanto o método **ROS** obteve 7,75 de DCG. Isso significa que apesar de terem classificado corretamente a mesma quantidade de amostras, o método **ROS** foi capaz de melhorar o conjunto de amostras para a construção de modelos de predição que identificam probabilisticamente mais famílias em situação de alta vulnerabilidade social.

Tabela 27: Resultados parciais do classificador AODE para o Dataset D_04 da Database 02

	AODE			
Método	**Precision**	**Recall**	**F-measure**	**DCG**
ROS	0,36 ±0,03	0,46 ±0,06	0,40 ±0,02	7,57 ±1,05
SPIDER2	0,37 ±0,03	0,45 ±0,03	0,41 ±0,03	3,82 ±3,71

Fonte: Autoria própria.

Para avaliar os resultados através das duas medidas *Recall* e DCG foi feita a normalização dos valores do DCG e posteriormente plotados em gráficos para facilitar a visualização dos resultados. A normalização $min - max$ dos valores foi feita na escala entre 0 (zero) e 1 (um) através da Equação 24:

$$(Z_i^k)_N = \frac{Z_i^k - Z_{min}^k}{Z_{max}^k - Z_{min}^k} \tag{24}$$

Os resultados combinados dessas duas medidas foram apresentados em gráficos onde o eixo **x** representa o valor do *Recall* da classe de interesse minoritária e o eixo **y** representa o valor do DCG. Estes gráficos estão apresentados nas Figuras 45, 46, 47 e 48.

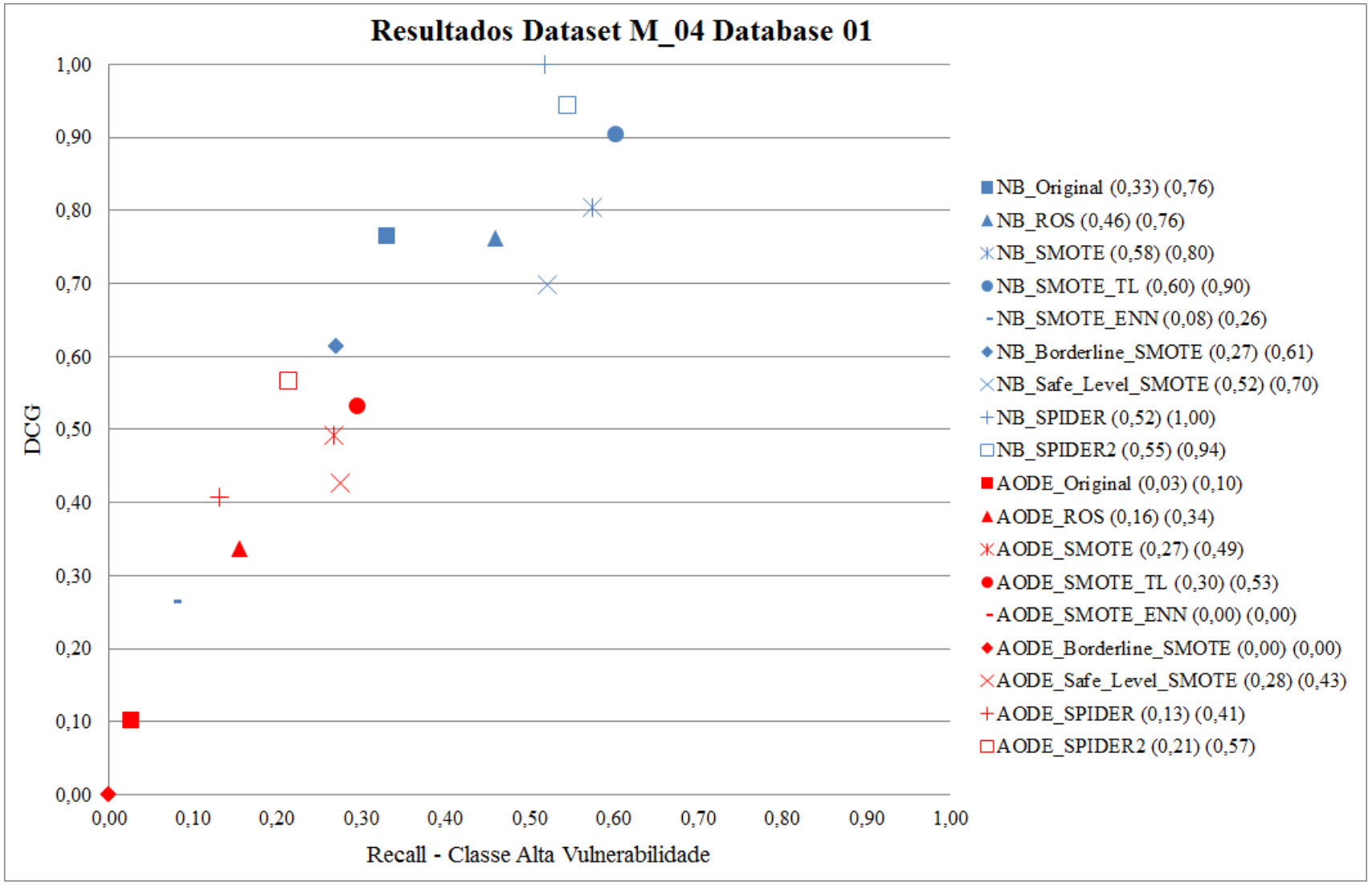


Figura 45: Gráfico Recall vs DCG Dataset M_04 Database 01

Fonte: Autoria própria.

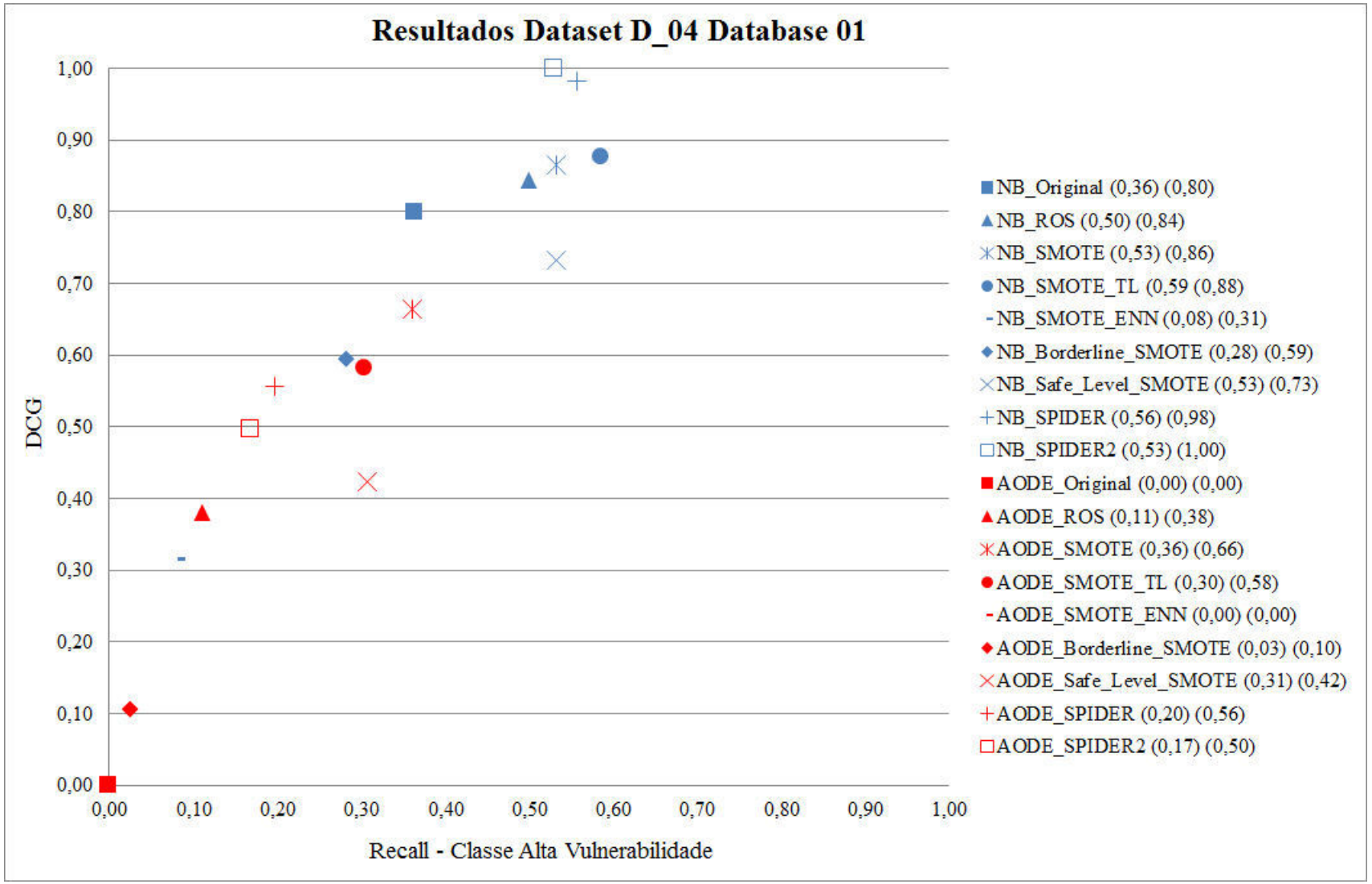


Figura 46: Gráfico Recall vs DCG Dataset D_04 Database 01

Fonte: Autoria própria.

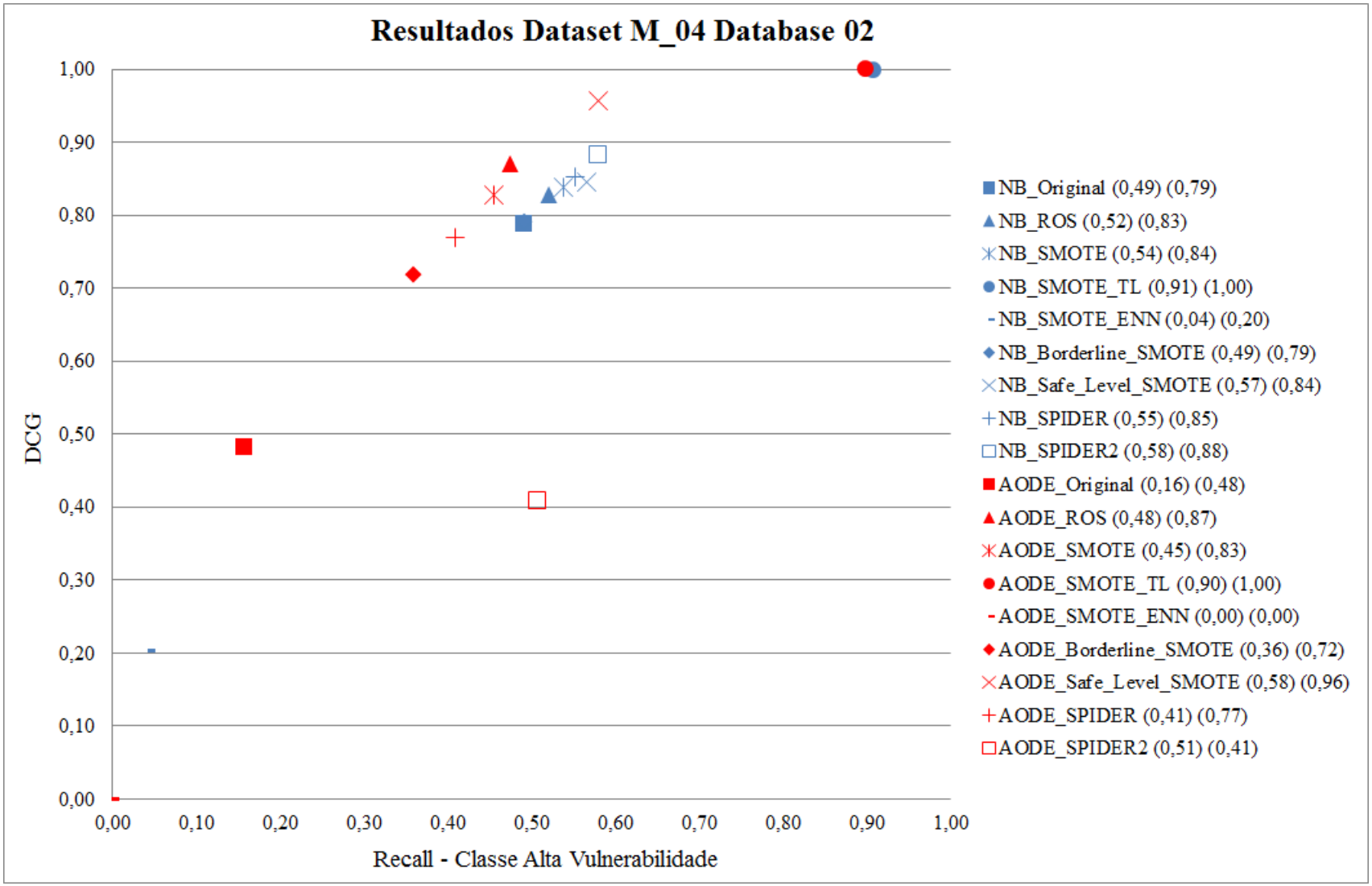


Figura 47: Gráfico Recall vs DCG Dataset M_04 Database 02

Fonte: Autoria própria.

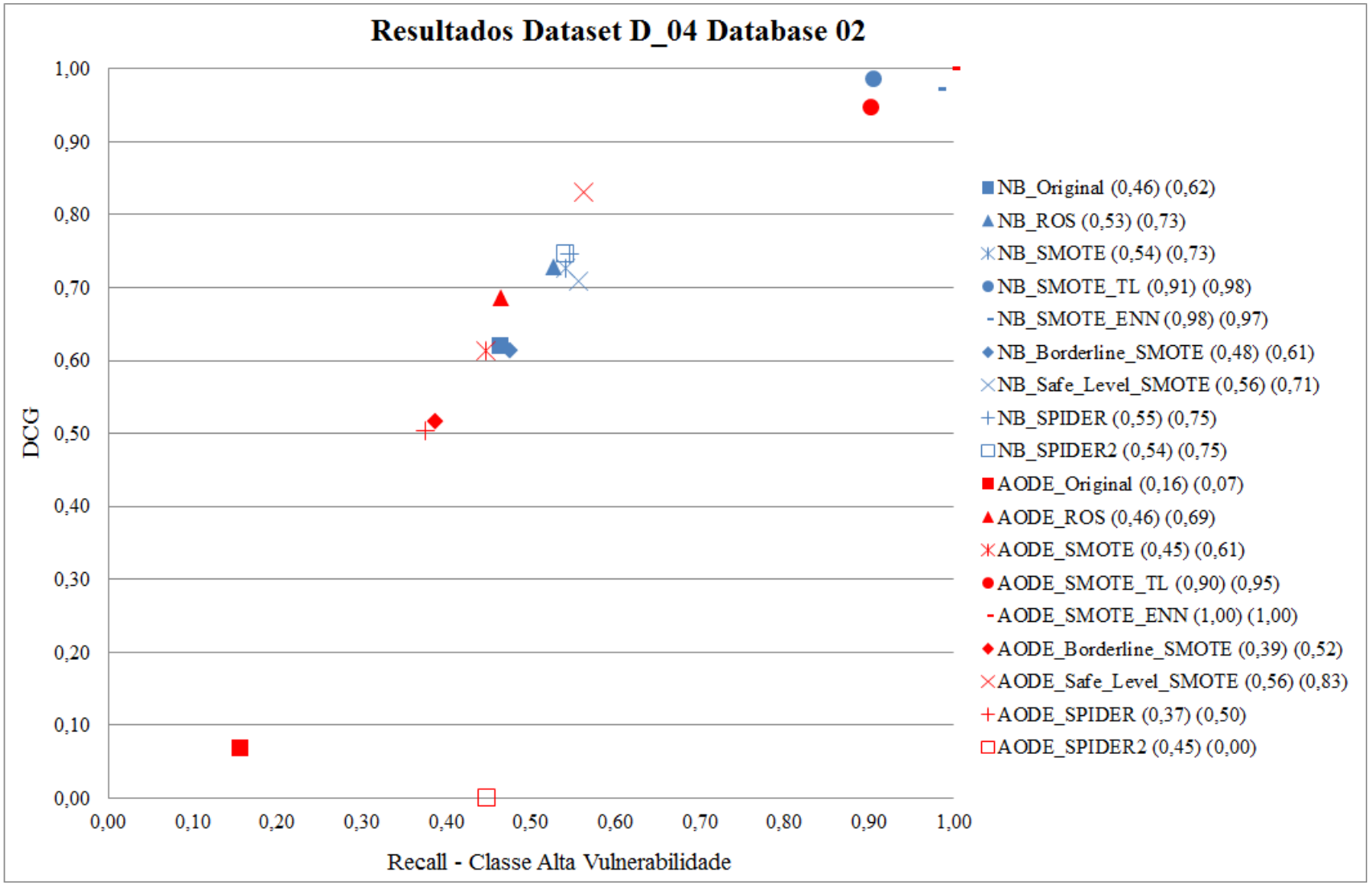


Figura 48: Gráfico Recall vs DCG Dataset D_04 Database 02

Fonte: Autoria própria.

7.3.7 AVALIAÇÃO DOS RESULTADOS

Os resultados foram analisados com base nos valores Recall da classe de interesse (Alta Vulnerabilidade) e DCG. Quanto maior o valor do Recall maior é a quantidade de famílias em situação de Alta Vulnerabilidade classificadas corretamente. Quanto maior o valor do DCG melhor foi o posicionamento das famílias em situação de Alta Vulnerabilidade na lista ordenada pela probabilidade da classe de Alta Vulnerabilidade. Os valores do DCG foram normalizados através da Equação 24 e variam entre 0 e 1.

Ao analisar os resultados do conjunto de dados **M_04** da **Database 01**, apresentados no gráfico da Figura 45, é possível observar que o classificador Naive Bayes obteve claramente melhores resultados do que o classificador AODE em todos os métodos de balanceamento utilizados. O melhor resultado para este conjunto de dados foi obtido pelo método de balanceamento SMOTE_TL com o classificador Naive Bayes onde o *Recall* foi de 60% e o DCG 0,90. Em comparação o conjunto de dados original sem aplicação de nenhum método de balanceamento obteve um *Recall* de 33% e DCG de 0,76 com o classificador Naive Bayes.

O melhor resultado obtido pelo classificador AODE também foi com o método de balanceamento SMOTE_TL obtendo um *Recall* de 30% e DCG de 0,53. Em comparação o conjunto de dados original sem aplicação de nenhum método de balanceamento com o classificador AODE obteve um *Recall* de 3% e DCG de 0,10.

O pior resultado obtido pelo classificador Naive Bayes foi com o método de balanceamento SMOTE_ENN com 0,08% de *Recall* e 0,26 de DCG. Já para o classificador AODE os piores resultados foram 0% de *Recall* e 0 de DCG obtido pelos métodos de balanceamento SMOTE_ENN e Borderline_SMOTE.

O método de balanceamento randômico ROS utilizado como *baseline* apresentou melhoras nos resultados para os dois classificadores quando comparado ao Dataset Original. Para o classificador Naive Bayes o Dataset Original obteve 0,33% de *Recall* e DCG de 0,76, já o método ROS obteve 0,46% de *Recall* e manteve o DCG de 0,76. Para o classificador AODE o Dataset Original obteve 0,03% de *Recall* e DCG de 0,10, já o método ROS obteve 0,16% de *Recall* e DCG de 0,34.

Considerando o Dataset M_04 da Database 01, o melhor resultado foi obtido pelo classificador Naive Bayes com o método de balanceamento SMOTE_TL, onde o *Recall* foi de 60% e o DCG 0,90. Isso implica em uma melhora de 27% no *Recall* em relação ao dataset Original. Neste cenário, seria possível identificar 21,6 famílias em alta vulnerabilidade e risco social que estariam invisíveis de um total de 36 famílias em alta vulnerabilidade existentes no conjunto de

teste.

Ao analisar os resultados do conjunto de dados **D_04** da **Database 01**, apresentados no gráfico da Figura 46, é possível observar que novamente o classificador Naive Bayes obteve claramente melhores resultados do que o classificador AODE em todos os métodos de balanceamento utilizados. Os melhores resultados para este conjunto de dados foram obtidos pelos métodos de balanceamento SMOTE_TL, SPIDER e SPIDER2 com o classificador Naive Bayes. Esses três métodos de balanceamento obtiveram resultados muito próximos. O SMOTE_TL obteve 59% de *Recall* e 0,88 de DCG. O SPIDER obteve 56% de *Recall* e 0,98 de DCG. O SPIDER2 obteve 53% de *Recall* e 1,00 de DCG. Em comparação o conjunto de dados original sem aplicação de nenhum método de balanceamento obteve um *Recall* de 36% e DCG de 0,80 com o classificador Naive Bayes.

O melhor resultado obtido pelo classificador AODE foi com o método de balanceamento SMOTE obtendo um *Recall* de 36% e 0,66 de DCG. Em comparação o conjunto de dados original sem aplicação de nenhum método de balanceamento obteve um *Recall* de 0% e DCG de 0 com o este mesmo classificador.

Os piores resultados foram 0% de *Recall* e 0 de DCG obtido pelo método de balanceamento SMOTE_ENN para os dois classificadores. Para o classificador AODE esse resultado também se repetiu com o conjunto de dados original sem nenhum balanceamento.

O método de balanceamento randômico ROS utilizado como *baseline* apresentou melhoras nos resultados para os dois classificadores quando comparado ao Dataset Original. Para o classificador Naive Bayes o Dataset Original obteve 0,36% de *Recall* e DCG de 0,80, já o método ROS obteve 0,50% de *Recall* e DCG de 0,84. Para o classificador AODE o Dataset Original obteve 0% de *Recall* e DCG de 0, já o método ROS obteve 0,11% de *Recall* e DCG de 0,38.

Considerando o Dataset D_04 da Database 01, o melhor resultado foi obtido pelo classificador Naive Bayes com o método de balanceamento SMOTE_TL, onde o *Recall* foi de 59% e o DCG 0,88. Isso implica em uma melhora de 23% no *Recall* em relação ao dataset Original. Neste cenário, seria possível identificar 21,24 famílias em alta vulnerabilidade e risco social que estariam invisíveis de um total de 36 famílias em alta vulnerabilidade existentes no conjunto de teste.

Ao analisar os resultados do conjunto de dados **M_04** da **Database 02**, apresentados no

gráfico da Figura 47, é possível observar que nos experimentos realizados com a Database 02 que contém mais amostras do que a Database 01 os resultados dos classificadores Naive Bayes e AODE foram similares, diferentemente dos resultados dos experimentos realizados com a Database 01 onde o classificador Naive Bayes foi superior em todos os resultados.

Os melhores resultados deste experimento foram alcançados pelos classificadores com o método de balanceamento SMOTE_TL, o classificador Naive Bayes obteve 91% de Recall e 1,00 de DCG, enquanto o AODE obteve 90% de Recall e 1,00 de DCG. Em comparação o conjunto de dados original sem aplicação de nenhum método de balanceamento obteve com o classificador Naive Bayes um *Recall* de 49% e DCG de 0,79 já com o classificador AODE o *Recall* obtido foi de 16% e DCG de 0,48.

Os piores resultados para os dois classificadores foram obtidos pelo método de balanceamento SMOTE_ENN onde para o classificador Naive Bayes foi obtido 0,04% de *Recall* e 0,20 de DCG enquanto para o classificador AODE foi obtido 0% de *Recall* e 0 de DCG.

O método de balanceamento randômico ROS utilizado como *baseline* apresentou melhoras nos resultados para os dois classificadores quando comparado ao Dataset Original. Para o classificador Naive Bayes o Dataset Original obteve 0,49% de *Recall* e DCG de 0,79, já o método ROS obteve 0,52% de *Recall* e DCG de 0,83. Para o classificador AODE o Dataset Original obteve 0,16% de *Recall* e DCG de 0,48, já o método ROS obteve 0,48% de *Recall* e DCG de 0,87.

Considerando o Dataset M_04 da Database 02, o melhor resultado foi obtido pelo classificador Naive Bayes com o método de balanceamento SMOTE_TL, onde o *Recall* foi de 91% e o DCG 1. Isso implica em uma melhora de 42% no *Recall* em relação ao dataset Original. Neste cenário, seria possível identificar 232,96 famílias em alta vulnerabilidade e risco social que estariam invisíveis de um total de 286 famílias em alta vulnerabilidade existentes no conjunto de teste.

Ao analisar os resultados do conjunto de dados **D_04** da **Database 02**, apresentados no gráfico da Figura 48, é possível observar que nos experimentos realizados os resultados dos classificadores foram similares entre si para alguns métodos de balanceamento, diferentemente dos resultados obtidos nos experimentos com a Database 01 onde o classificador Naive Bayes foi superior em todos os resultados. Ao contrário do experimento anterior com o conjunto de dados M_04 da Database 02 onde o método SMOTE_TL obteve os melhores resultados, neste experimento os melhores resultados foram obtidos pelo método de balanceamento SMOTE_ENN que curiosamente obteve os piores resultados no experimento anterior. Neste caso os resultados

obtidos pelo método de balanceamento SMOTE_ENN com o classificador Naive Bayes foi de 98% de *Recall* e DCG de 0,97, já o classificador AODE obteve 100% de Recall e DCG de 1,00. Em comparação o conjunto de dados original sem aplicação de nenhum método de balanceamento obteve com o classificador Naive Bayes um *Recall* de 46% e DCG de 0,62, já com o classificador AODE o *Recall* obtido foi de 16% e DCG de 0,07.

Diferentemente dos piores resultados dos experimentos anteriores, neste experimento nenhum método obteve valor 0 nos seus resultados. Os piores resultados foram obtidos pelo classificador AODE com o conjunto de dados original atingindo 16% de *Recall* e 0,07 de DCG e com o método de balanceamento SPIDER2 atingindo 45% de *Recall* e 0 de DCG.

O método de balanceamento randômico ROS utilizado como *baseline* apresentou melhoras nos resultados para os dois classificadores quando comparado ao Dataset Original. Para o classificador Naive Bayes o Dataset Original obteve 0,46% de *Recall* e DCG de 0,62, já o método ROS obteve 0,53% de *Recall* e DCG de 0,73. Para o classificador AODE o Dataset Original obteve 0,16% de *Recall* e DCG de 0,07, já o método ROS obteve 0,46% de *Recall* e DCG de 0,69.

Considerando o Dataset D_04 da Database 02, o melhor resultado foi obtido pelo classificador AODE com o método de balanceamento SMOTE_ENN, onde o *Recall* foi de 100% e o DCG 1. Isso implica em uma melhora de 99,84% no *Recall* em relação ao dataset Original. Neste cenário, seria possível identificar todas as 286 famílias em alta vulnerabilidade e risco social existentes no conjunto de teste.

Através da **avaliação geral** dos resultados plotados nos gráficos foi possível identificar que o método de balanceamento **SMOTE_TL** obteve os melhores resultados em todos os cenários.

Na Database 01 o classificador Naive Bayes obteve resultados claramente melhores que o AODE e os 4 métodos de balanceamento que tiveram os melhores resultados em ambos os conjuntos de dados foram **SPIDER**, **SPIDER2**, **SMOTE** e **SMOTE_TL**.

Na Database 02 os resultados do classificador AODE melhoraram muito em relação aos resultados da Database 01. Neste caso, os resultados do AODE e do Naive Bayes foram parecidos. Em ambos os conjuntos de dados o método de balanceamento **SMOTE_TL** obteve ótimos resultados com os dois classificadores utilizados. No conjunto de dados D_04 o método SMOTE_ENN se destacou obtendo o melhor resultado entre todos.

Durante o estudo dos métodos de balanceamento foi possível observar que os métodos

propostos pelos seus respectivos autores ao longo do tempo eram evoluções do método SMOTE na tentativa de melhorar os resultados através da eliminação do *overfitting*. Durante as análises dos resultados nem sempre os métodos mais atuais foram capazes de obterem melhores resultados quando comparado ao SMOTE. Apenas o método SMOTE_TL obteve melhores resultados do que o SMOTE em todos os casos. Os demais métodos não mantiveram esse padrão sendo que em determinados momentos seus resultados foram melhores do que o SMOTE mas em outros momentos o resultado foi muito inferior.

As Tabelas 28 e 29 apresentam os resultados obtidos pelos conjuntos de dados original sem balanceamento e pelos métodos que obtiveram os melhores resultados em cada uma das configurações com base no Recall da classe de interesse e no DCG. Na última coluna dessas tabelas são apresentados os valores percentuais de melhora do Recall da classe de interesse obtido pelo melhor método de balanceamento quando comparado com o resultado obtido pelo conjunto de dados original sem balanceamento.

Tabela 28: Melhores resultados Database 01

		Original		Melhor			
		Recall	DCG	Método	Recall	DCG	Melhora
Naive Bayes	D_	0,36	0,8	SMOTE_TL	0,59	0,88	23%
	M_	0,33	0,76	SMOTE_TL	0,6	0,9	27%
AODE	D_	0	0	SMOTE	0,36	0,66	36%
	M_	0,03	0,1	SMOTE_TL	0,3	0,53	30%

Fonte: Autoria própria.

Tabela 29: Melhores resultados Database 02

		Original		Melhor			
		Recall	DCG	Método	Recall	DCG	Melhora
Naive Bayes	D_	0,46	0,62	SMOTE_ENN	0,98	0,97	52%
	M_	0,49	0,79	SMOTE_TL	0,91	1	42%
AODE	D_	0,16	0,07	SMOTE_ENN	1	1	84%
	M_	0,16	0,48	SMOTE_TL	0,9	1	74%

Fonte: Autoria própria.

7.4 ANÁLISE DOS EXPERIMENTOS

Através da avaliação dos resultados dos experimentos utilizando métodos de balanceamento para deixar os conjuntos de dados com a mesma distribuição de amostras entre as classes, foi possível verificar que alguns métodos de balanceamento foram capazes de melhorar os resultados dos classificadores porém alguns métodos tiveram efeito contrário prejudicando o resultado em relação ao conjunto de dados original.

Ao tentar realizar a análise dos resultados através das métricas padrões foi percebido que os métodos de balanceamento, mesmo aumentando os valores dessas medidas padrões não estavam produzindo bons resultados em alguns casos. Em função deste problema buscou-se outras formas para avaliar os resultados para conseguir identificar quais métodos haviam de fato beneficiado as predições feitas pelos classificadores. A nova forma encontrada para avaliar os resultados foi através da utilização da medida *Recall* em conjunto com uma medida de qualidade de *ranking* chamada DCG.

Utilizando as duas medidas *Recall* e DCG para avaliar os resultados, o método de balanceamento SMOTE_TL se mostrou o melhor entre os métodos testados quando avaliado todos os cenários. Já o método SMOTE_ENN que apresentou desempenho ruim em alguns cenários conseguiu obter ótimos resultados quando utilizado com o conjunto de dados D_04 da Database 02.

Em todos os cenários avaliados o método de balanceamento randômico ROS utilizado como *baseline* obteve melhores resultados quando comparado aos resultados obtidos quando utilizado o Dataset Original onde as amostras não estão balanceadas. Estes resultados indicam que o desbalanceamento influencia negativamente o desempenho dos classificadores utilizados neste trabalho para tentar resolver o problema proposto, de modo que, ao realizar um balanceamento randômico os resultados obtidos foram melhorados.

Avaliando o desempenho dos classificadores, de modo geral, o Naive Bayes foi superior ao AODE. Com a Database 01 o Naive Bayes obteve melhores resultados do que o AODE com todos os métodos de balanceamento utilizados e inclusive com o dataset Original sem balanceamento. Com o Database 02, apesar dos resultados obtidos pelo AODE terem melhorado, o classificador Naive Bayes continuou obtendo melhores resultados em quase todos os casos, sendo que o AODE conseguiu um resultado pouco superior apenas com o método de balanceamento SMOTE_ENN no Dataset D_04.

Ao avaliar os resultados com base nas duas abordagens utilizadas para reduzir os conjuntos de dados a apenas duas classes é possível observar que de modo geral os resultados

foram bastante parecidos para os datasets onde a classe Baixa e Média vulnerabilidade foram combinada (*merged*) e para os datasets onde a classe Baixa vulnerabilidade foi descartada. O fato de ter descartado ou mantido as amostras da classe Baixa Vulnerabilidade não afetou de modo geral os resultados. Apenas no caso da Database 02 com o método de balanceamento SMOTE_ENN houve uma mudança significativa nos resultados, onde o Dataset M_04 obteve resultados ruins e o Dataset D_04 obteve bons resultados. No caso do Dataset M_04 com o classificador Naive Bayes obteve-se 0,04% de *Recall* e 0,20 de DCG enquanto o classificador AODE obteve 0% de *Recall* e 0 de DCG. Porém para o Dataset D_04 os resultados foram bem diferentes, neste caso, o classificador Naive Bayes apresentou 0,98% de *Recall* e 0,97 de DCG, enquanto o classificador AODE obteve 100% de *Recall* e 1 de DCG.

8 CONSIDERAÇÕES FINAIS

A Busca Ativa das famílias em situação de vulnerabilidade social é muito importante para que o Estado possa assistir as famílias mais necessitadas que por algum motivo não procuram de forma espontânea os serviços da assistência social. Atualmente não existe nenhuma ferramenta específica para realização da Busca Ativa das famílias para os municípios.

O presente trabalho buscou estudar e aplicar técnicas de mineração de dados para viabilizar a construção de uma ferramenta que auxilie na Busca Ativa dessas famílias.

Neste trabalho foram investigadas e utilizadas algumas técnicas de mineração de dados para auxiliar a identificação das famílias em situação de vulnerabilidade e risco social, através de informações coletadas pela equipe da assistência social, no intuito de facilitar e auxiliar a Busca Ativa das famílias.

Como resultado pretendia-se obter uma técnica de classificação satisfatória para auxiliar os municípios a realizarem a classificação de vulnerabilidade das famílias e auxiliar no processo de Busca Ativa das famílias em situação de vulnerabilidade e risco social através da classificação obtida pelo modelo de predição.

Após analisar os resultados dos experimentos preliminares realizados foi possível observar que os modelos utilizados sempre classificavam a amostra como membro da classe majoritária em função do conjunto de treinamento possuir amostras desbalanceadas. Após realizar um balanceamento das classes de forma manual, reduzindo o número de amostras por classe, foram realizados experimentos adicionais e os resultados, apesar de terem apresentado uma taxa de acerto menor em comparação ao conjunto desbalanceado, apresentou um resultado positivo, pois o classificador passou a predizer mais corretamente as amostras balanceadas.

Também foi realizada uma análise detalhada das informações contidas nas amostras que os algoritmos estavam classificando de forma errada. Analisando, por exemplo, uma família que obteve a classificação como “Alta Vulnerabilidade” segundo o formulário de avaliação de vulnerabilidade e risco social, e foi classificada como sendo “Média Vulnerabilidade” pelo algoritmo, pôde-se observar de forma empírica que as informações que fizeram essa família ser

classificada como “Alta Vulnerabilidade” pelo formulário são referentes às questões específicas que só existem para as famílias que tiveram o formulário preenchido. Para esses casos o algoritmo não irá conseguir identificar que a família é de “Alta Vulnerabilidade”, pois os atributos comuns de todas as famílias não possuem essas informações.

Diante desses fatos mostrou-se necessário aplicar técnicas de balanceamento no conjunto de treinamento a fim de obter melhores resultados. Também se mostrou importante que o algoritmo utilizado para a classificação pudesse não apenas classificar a amostra em um dos rótulos de vulnerabilidade, mas que também fosse capaz de aferir o grau de probabilidade da família estar em situação de vulnerabilidade; uma vez que o intuito final da solução é identificar quais famílias precisam ser prioritariamente assistidas. Assim, ao invés de obter todas as famílias rotuladas em uma das classes de vulnerabilidade, será obtida uma estimativa de probabilidade das famílias mais vulneráveis. Com essa estimativa é possível fornecer uma lista ordenada de prioridade de atendimento das famílias com base na probabilidade de vulnerabilidade auxiliando no processo de Busca Ativa.

O grau de probabilidade de vulnerabilidade das famílias também se faz necessário devido à capacidade de atendimento dos serviços. Caso tenha-se, por exemplo, um volume grande de famílias classificadas como “Alta Vulnerabilidade” os serviços podem não ter capacidade para atender todas ao mesmo tempo sendo necessário gerar uma ordem prioritária para o atendimento.

Novos experimentos foram realizados após a aplicação de métodos de balanceamento nos conjuntos de dados. Para os novos experimentos foi realizada uma nova extração de informações da base de dados do sistema IRSAS de modo que os experimentos foram realizados utilizando a Database 01 que foi utilizada nos experimentos preliminares e uma nova Database 02 que foi utilizada nos experimentos finais.

Os novos experimentos mostraram bons resultados pois após a utilização dos métodos de balanceamento nos conjuntos de dados, os valores das medidas *Recall*, Precisão e F-measure melhoraram consideravelmente em alguns casos. Nem sempre os métodos de balanceamento foram capazes de melhorar os resultados, pelo contrário, em alguns casos os resultados obtidos foram piores do que quando utilizado o conjunto de dados original com as classes desbalanceadas.

Ao analisar de forma mais detalhada os resultados obtidos foi possível identificar que comparando os resultados através das medidas *Recall*, Precisão ou F-measure, não foi possível identificar qual método se mostrava melhor para resolver o problema proposto. Diante deste fato novas formas para avaliar os resultados foram propostas. A aplicação da medida DCG

tornou possível avaliar a qualidade do resultado de cada método com base em uma lista gerada por cada classificador, contendo as amostras classificadas e ordenadas com base no valor da probabilidade dessa amostra pertencer a classe Alta Vulnerabilidade. Utilizando a medida DCG juntamente com a medida *Recall*, foi possível comparar os resultados de cada método e identificar aqueles que produziram melhores resultados para o problema real tratado por esse trabalho.

Por fim, ao avaliar os cenários de forma geral, o método de balanceamento SMOTE_TL foi o que produziu melhores resultados tanto com o classificador Naive Bayes quanto com o AODE nas duas bases de dados testadas.

8.1 TRABALHOS FUTUROS

Como trabalho futuro para este tema pode-se investigar e utilizar novas abordagens para resolução do problema proposto como, por exemplo, a utilização de métodos de classificação preparados para lidar com o problema do desbalanceamento diretamente na etapa de construção do modelo de predição (como citado na Seção 4), diferentemente das abordagens utilizadas neste trabalho, onde foram utilizadas técnicas de balanceamento do conjunto de dados com métodos de classificação padrões.

Também como trabalho futuro pode-se estudar e aplicar outros métodos Bayesianos que tentam aprender a estrutura da rede a fim de buscar melhores resultados nas predições das famílias em situação de alta vulnerabilidade social.

Outro trabalho futuro pode ser a validação e aplicação real dos resultados obtidos na busca ativa das famílias em situação de vulnerabilidade. Este trabalho pode ser realizado através da aplicação do modelo de predição em 100% das famílias do município de Cascavel existentes na base de dados do sistema IRSAS. Após a aplicação do modelo será obtida uma lista ordenada das famílias em situação de vulnerabilidade e risco social. Essa lista das famílias poderá ser entregue para a equipe de técnicos da secretaria de assistência social do município que então irá visitar as famílias apontadas com maior probabilidade de estarem em alta vulnerabilidade social, verificando se a predição realizada está correta e se a família realmente se encontra na situação indicada pelo modelo de predição. Desse modo poderá ser realizada uma aplicação real e prática da técnica estudada a fim de verificar se o modelo é capaz de apoiar o processo da busca ativa das famílias em situação de vulnerabilidade e risco social.

REFERÊNCIAS

ALCALA-FDEZ, J. et al. Keel data-mining software tool: Data set repository and integration of algorithms and experimental analysis framework. **Journal of Multiple-Valued Logic and Soft Computing**, v. 17, n. 2-3, p. 255–287, 2011.

ALCALA-FDEZ, J. et al. Keel: a software tool to assess evolutionary algorithms for data mining problems. **Soft Computing**, Springer-Verlag, v. 13, n. 3, p. 307–318, 2009.

BATISTA, G.; PRATI, R.; MONARD, M. A study of the behavior of several methods for balancing machine learning training data. **SIGKDD Explorations**, v. 6, n. 1, p. 20–29, 2004.

BRASIL. **Ministério do Desenvolvimento Social e Combate à Fome. Orientações Técnicas: Centro de Referência de Assistência Social - CRAS**. Brasília: , 2009.

BRASIL. **Busca Ativa: O que é a busca ativa do plano Brasil sem miséria**. 2013. Disponível em: `http://www.mds.gov.br/falemds/perguntas-frequentes/superacao-da-extrema-pobreza\%20/plano-brasil-sem-miseria-1/busca-ativa/`. Acesso em 20 jun. 2015.

BUNKHUMPORNPAT, C.; SINAPIROMSARAN, K.; LURSINSAP, C. Safe-level-smote: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem. In: **Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining**. : Springer-Verlag, 2009. (Lecture Notes on Computer Science, v. 5476), p. 475–482.

CHAWLA, N. et al. Smote: Synthetic minority over-sampling technique. **Journal of Artificial Intelligence Research**, v. 16, p. 321–357, 2002.

CHICKERING, D. Learning bayesian networks is np-complete. In: FISHER, D.; LENZ, H.-J. (Ed.). **Learning from Data**. : Springer New York, 1996, (Lecture Notes in Statistics, v. 112). p. 121–130.

COHEN, G. et al. Learning from imbalanced data in surveillance of nosocomial infection. **Artificial Intelligence in Medicine**, Elsevier, v. 37, n. 1, p. 7–18, 2006.

DOUGHERTY, G. **Pattern Recognition and Classification: An Introduction**. : Springer, 2013.

FAWCETT, T.; PROVOST, F. Adaptive fraud detection. **Data mining and knowledge discovery**, Springer, v. 1, n. 3, p. 291–316, 1997.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. **AI Magazine**, v. 17, p. 37–54, 1996.

FERNÁNDEZ, A. et al. A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets. **Fuzzy Sets and Systems**, Elsevier North-Holland, Inc., Amsterdam, The Netherlands, v. 159, n. 18, p. 2378–2398, set. 2008.

FLORES, M. J. et al. Handling numeric attributes when comparing bayesian network classifiers: Does the discretization method matter? **Applied Intelligence**, Kluwer Academic Publishers, Hingham, MA, USA, v. 34, n. 3, p. 372–385, 2011.

FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian network classifiers. **Machine Learning**, Kluwer Academic Publishers, v. 29, n. 2-3, p. 131–163, 1997.

GALAR, M. et al. A review on ensembles for the class imbalance problem: Bagging-, boosting-, and hybrid-based approaches. **IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews**, v. 42, n. 4, p. 463–484, 2012.

GALAR, M. et al. Eusboost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling. **Pattern Recognition**, v. 46, n. 12, p. 3460 – 3471, 2013.

GRZYMALA-BUSSE, J. W.; STEFANOWSKI, J.; WILK, S. A comparison of two approaches to data mining from imbalanced data. **Journal of Intelligent Manufacturing**, Springer, v. 16, n. 6, p. 565–573, 2005.

HALL, M. et al. The weka data mining software: An update. **SIGKDD Explorations Newsletter**, ACM, New York, NY, USA, v. 11, n. 1, p. 10–18, nov. 2009.

HAN, H.; WANG, W.; MAO, B. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: **Proceedings of the International Conference on Intelligent Computing**. : Springer-Verlag, 2005. (Lecture Notes on Computer Science, v. 3644), p. 878–887.

HILAS, C. S.; MASTOROCOSTAS, P. A. An application of supervised and unsupervised learning approaches to telecommunications fraud detection. **Knowledge-Based Systems**, Elsevier, v. 21, n. 7, p. 721–726, 2008.

HU, S. et al. Msmote: Improving classification performance when training data is imbalanced. In: **Proceedings of the Second International Workshop on Computer Science and Engineering**. 2009. v. 2, p. 13–17.

JIANG, L. et al. Survey of improving naive bayes for classification. In: **Proceedings of the 3rd International Conference on Advanced Data Mining and Applications**. Berlin, Heidelberg: Springer-Verlag, 2007. p. 134–145.

JIANG, L. et al. Evolutional naive bayes. **Proceedings of the International Symposium on Intelligent Computation and its Application**, p. 344–350, 2005.

JÄRVELIN, K.; KEKÄLÄINEN, J. Cumulated gain-based evaluation of ir techniques. **ACM Transactions on Information Systems**, ACM, New York, NY, USA, v. 20, n. 4, p. 422–446, 2002.

KEOGH, E. J.; PAZZANI, M. J. Learning augmented bayesian classifiers: A comparison of distribution-based and classification-based approaches. **In Proceedings of the International Workshop on Artificial Intelligence and Statistics**, p. 225–230, 1999.

KOHAVI, R.; PROVOST, F. Glossary of terms. **Machine Learning**, v. 30, n. 2-3, p. 271–274, 1998.

KUBAT, M.; HOLTE, R. C.; MATWIN, S. Machine learning for the detection of oil spills in satellite radar images. **Machine learning**, Springer, v. 30, n. 2-3, p. 195–215, 1998.

LAURIKKALA, J. **Improving Identification of Difficult Small Classes by Balancing Class Distribution. Tech. Report A-2001-2**. 2001.

LÓPEZ, V. et al. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. **Information Sciences**, v. 250, p. 113 – 141, 2013.

LÓPEZ, V. et al. Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics. **Expert Systems with Applications**, v. 39, n. 7, p. 6585 – 6608, 2012.

LÓPEZ, V. et al. Addressing imbalanced classification with instance generation techniques: Ipade-id. **Neurocomputing**, v. 126, n. 0, p. 15 – 28, 2014.

LUENGO, J. et al. Addressing data complexity for imbalanced data sets: analysis of smote-based oversampling and evolutionary undersampling. **Soft Computing**, Springer-Verlag, v. 15, n. 10, p. 1909–1936, 2011.

MAHESHWARI, S.; AGRAWAL, J.; SHARMA, S. New approach for classification of highly imbalanced datasets using evolutionary algorithms. **International Journal of Scientific & Engineering Research**, Citeseer, v. 2, n. 7, p. 1–5, 2011.

MONARD, M. C.; BARANAUSKAS, J. A. **Conceitos Sobre Aprendizado de Máquina. In: Solange O. Rezende. (Org.). Sistemas Inteligentes Fundamentos e Aplicações**. 1. ed. Barueri - SP: Manole Ltda, 2003. 89-114 p.

NAPIERALA, K.; STEFANOWSKI, J.; WILK, S. Learning from imbalanced data in presence of noisy and borderline examples. In: SPRINGER. **Proceedings of the 7th International Conference on Rough Sets and Current Trends in Computing**. 2010. p. 158–167.

PENA, S. D. **Thomas Bayes, o "Cara"!** Rio de Janeiro: Ciência Hoje, 2006. 22–29 p.

RAMENTOL, E. et al. Smote-rsb*: A hybrid preprocessing approach based on oversampling and undersampling for high imbalanced data-sets using smote and rough sets theory. **Knowledge and Information Systems**, v. 33, n. 2, p. 245–265, 2012.

SAHAMI, M. Learning limited dependence bayesian classifiers. In: **KDD**. 1996. v. 96, p. 335–338.

SALAMA, G.; ABDELHALIM, M.; ZEID, M. Experimental comparison of classifiers for breast cancer diagnosis. In: **Proceedings of the Seventh International Conference on Computer Engineering Systems**. 2012. p. 180–185.

SEBASTIANI, P.; ABAD, M.; RAMONI, M. Bayesian networks. In: MAIMON, O.; ROKACH, L. (Ed.). **Data Mining and Knowledge Discovery Handbook**. : Springer US, 2010. p. 175–208.

STEFANOWSKI, J.; WILK, S. Selective pre-processing of imbalanced data for improving classification performance. In: **Proceedings of the 10th International Conference in Data Warehousing and Knowledge Discovery**. : Springer, 2008. (Lecture Notes on Computer Science, v. 5182), p. 283–292.

TAN, P. N.; STEINBACH, M.; KUMAR, V. **Introdução ao Data Mining - Mineração de Dados**. Rio de Janeiro: Ciência Moderna, 2009.

TERRIN, M. A. P.; Silla Jr., C. N.; BUGATTI, P. H. Utilizando técnicas de mineração de dados para apoiar a busca ativa de famílias em situação de vulnerabilidade e risco social. **XLI Seminário Integrado de Software e Hardware (SEMISH)**, Brasília, Brasil, p. 1051–1062, 2014.

TOMEK, I. Two Modifications of CNN. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 7(2), p. 679–772, 1976.

WEBB, G. I.; BOUGHTON, J. R.; WANG, Z. Not so naive bayes: Aggregating one-dependence estimators. **Machine Learning**, Kluwer Academic Publishers, v. 58, n. 1, p. 5–24, 2005.

WEISS, G. M.; PROVOST, F. Learning when training data are costly: The effect of class distribution on tree induction. **Journal of Artificial Intelligence Research**, AI Access Foundation, USA, v. 19, n. 1, p. 315–354, out. 2003.

WILSON, D.; MARTINEZ, T. Reduction techniques for instance-based learning algorithms. **Machine Learning**, Kluwer Academic Publishers, v. 38, n. 3, p. 257–286, 2000.

WILSON, D. L. Asymptotic properties of nearest neighbor rules using edited data. **IEEE Transactions on Systems, Man and Cybernetics**, IEEE, n. 3, p. 408–421, 1972.

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.

WOLPERT, D.; MACREADY, W. No free lunch theorems for optimization. **IEEE Transactions on Evolutionary Computation**, v. 1, n. 1, p. 67–82, Apr 1997.

ZHENG, Z.; WEBB, G. Lazy learning of bayesian rules. **Machine Learning**, Kluwer Academic Publishers, v. 41, n. 1, p. 53–84, 2000.

ZHENG, Z.; WU, X.; SRIHARI, R. Feature selection for text categorization on imbalanced data. **ACM Sigkdd Explorations Newsletter**, ACM, v. 6, n. 1, p. 80–89, 2004.