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

Uso de Informação de Histórico de Beneficiários para Regulação em Saúde

Karoline de Moura Farias

Teresina-PI, 10 de setembro de 2019

Karoline de Moura Farias

Uso de Informação de Histórico de Beneficiários para Regulação em Saúde

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

Universidade Federal do Piauí – UFPI

Centro de Ciências da Natureza

Programa de Pós-Graduação em Ciência da Computação

Orientador: André Macêdo Santana

Coorientador: Pedro de Alcântara dos Santos Neto

Teresina-PI

10 de setembro de 2019

Karoline de Moura Farias
Uso de Informação de Histórico de Beneficiários para Regulação em Saúde/ Karoline de Moura Farias. – Teresina-PI, 10 de setembro de 2019-
88 p. : il. (algumas color.) ; 30 cm.

Orientador: André Macêdo Santana

Dissertação (Mestrado) – Universidade Federal do Piauí – UFPI
Centro de Ciências da Natureza
Programa de Pós-Graduação em Ciência da Computação, 10 de setembro de 2019.

1. Regulação. 2. Aprendizagem de Máquina. 3. Histórico de Pacientes I. André Macêdo Santana. II. Universidade Federal do Piauí. III. Uso de Informação de Histórico de Beneficiários para Regulação em Saúde.

CDU 02:141:005.7

"Uso de Informação de Histórico de Beneficiários para Regulação em Saúde"

KAROLINE DE MOURA FARIAS

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

Aprovada por:

Prof. André Macedo Santana
(Presidente da Banca Examinadora)

Prof. João Paulo Pordeus Gomes
(Examinador Externo à Instituição)

Prof. Pedro de Alcântara dos Santos Neto
(Examinador Interno)

Prof. Vinicius Ponte Machado
(Examinador Interno)

Teresina, 10 de setembro de 2019

Aos meus pais, Antônio Luiz de Farias e Maria da Paz Moura Santos Farias, meus grandes super-heróis.

Agradecimentos

Primeiramente, agradeço a Deus por todas as vezes que foi meu confidente, trazendo-me paz e tranquilidade nos momentos mais difíceis.

Agradeço aos meus pais, Maria da Paz Moura Santos Farias e Antônio Luiz de Farias, por investirem na minha educação acreditando no poder de transformação que o conhecimento pode trazer à vida das pessoas.

Aos meus irmãos, por serem meus amigos e me apoiarem incondicionalmente.

Agradeço ao meu orientador, André Macêdo Santana, por todos os conselhos, pela paciência e ajuda desde os meus primeiros passos na graduação. Agradeço também ao meu coorientador, Pedro de Alcântara dos Santos Neto, por me acolher como orientanda e me auxiliar diretamente na realização deste trabalho.

Aos meus amigos pela força que me dão constantemente, além do apoio e motivação constantes para que eu continuasse firme e forte nesta caminhada de aprimoramento.

A todos os professores que contribuíram para a minha formação desde a infância até os dias de hoje, e por servirem de inspiração para meu crescimento como pessoa.

À Universidade Federal do Piauí pelas oportunidades oferecidas além do apoio ao desenvolvimento de pesquisas científicas.

"Ninguém está mais longe da verdade
do que aquele que sabe todas as respostas."
(Chuang Tsu)

Resumo

Regulação é um mecanismo de controle utilizado por uma Operadora de Plano de Saúde (OPS) com objetivo de minimizar o desperdício de recursos por meio da análise de solicitações realizadas pelos beneficiários. Normalmente as solicitações passam por uma avaliação administrativa que certifica se a solicitação cumpre parâmetros não-técnicos (carência, adimplência e outros). Além disso, passa por uma avaliação especializada com profissionais que ficam à disposição para análise das solicitações. Uma das estratégias utilizadas para otimizar essa tarefa é o uso de um sistema que automatize parte desse processo por meio da utilização de aprendizagem de máquina (AM). O objetivo deste trabalho é aperfeiçoar o processo de aprendizagem supervisionada da regulação automatizada acrescentando informação do histórico de regulação dos beneficiários. Este estudo se baseia na ideia de que o histórico de beneficiários fornece informações relevantes sobre o processo de regulação, e, que possíveis solicitações posteriores, sigam, de alguma forma, um padrão baseado em solicitações antigas. A metodologia proposta utiliza três tipos de representação de informações: representação binária, *term-frequency* (TF) e *term-frequency inverse document frequency* (TF-IDF). Para cada uma dessas representações, são aplicados algoritmos de seleção de atributos (*Consistency Subset Eval* (CSE), *Correlation Feature Subset* (CFS), *Wrapper Subset Eval* (WSE), Ganho de Informação (IG), Razão de Ganho (*Gain Ratio* - GR) e Relief) e tranformação de atributos (*Principal Component Analysis* (PCA), Kernel PCA, *Independent Component Analysis* (ICA) e *Latent Semantic Analysis* (LSA)) . Na etapa de aprendizagem supervisionada são testados os algoritmos classificação: *Random Forest* (RF), *Naive Bayes* (NB), *K-nearest neighbors* (KNN) e *Support Vector Machine* (SVM). Os resultados obtidos pelos classificadores são avaliados a partir das métricas acurácia, precisão, *recall*, índice Kappa e *precision-recall curve* (PRC). A partir dos resultados também é avaliado se houve melhora significativa ou não, comparando os resultados da aprendizagem com e sem o histórico. Essa avaliação é realizada utilizando o teste de hipótese Z. Os resultados apontam uma melhora significativa, em todos os classificadores testados, onde o melhor resultado obtido foi utilizando o classificador RF com representações TF e TF-IDF com seleção de atributos.

Palavras-chaves: Aprendizagem de Máquina, Histórico Beneficiários em Regulação, Regulação Médica em Planos de Saúde

Abstract

Prior Authorization is a control mechanism used by a Health Insurance Providers (HIP) to minimize the waste of resources through the analysis of requests made by beneficiaries. Normally, the applications go through an administrative evaluation that certifies if the application complies with non-technical parameters (lack of funds, compliance and others). In addition, it undergoes a specialized evaluation with professionals who are available to analyze the applications. One of the strategies used to optimize this task is the use of a system that automates part of this process through the use of machine learning (ML). The objective of this work is to improve the process of supervised learning of automated prior authorization by adding information from the history of beneficiaries. This study is based on the idea that the history of beneficiaries provides relevant information on the prior authorization process, and that possible subsequent requests somehow follow a pattern based on old requests. The proposed methodology uses three types of information representation: binary representation, term-frequency. (TF) and term-frequency inverse document frequency (TF-IDF). For each of these representations, feature selection algorithms are applied (Consistency Subset Eval (CSE), Correlation Feature Subset (CFS), Wrapper Subset Eval (WSE), Information Gain (IG), Gain Ratio (GR) and Relief) and feature transformation (Principal Component Analysis (PCA), Kernel PCA, Independent Component Analysis (ICA) and Latent Semantic Analysis (LSA)). In the supervised learning stage the classification algorithms are tested: Random Forest (RF), Naive Bayes (NB), K-nearest neighbors (KNN) and Support Vector Machine (SVM). The results obtained by the classifiers are evaluated from the metrics accuracy, precision, recall, Kappa index and precision-recall curve (PRC). From the results it is also evaluated if there was significant improvement or not, comparing the learning outcomes with and without the beneficiaries' historical. This evaluation is performed using the hypothesis test Z. The results show a significant improvement in all the classifiers tested, where the best result was obtained using the RF classifier with TF and TF-IDF representations with feature selection.

Keywords: Beneficiaries' Historical, Machine Learning, Prior Authorization in Health Insurance Providers.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

A	Acurácia
AB	AdaBoost
AM	Aprendizagem de Máquina
ANS	Agência Nacional de Saúde Suplementar
BD	Banco de Dados
BF	Best First
BIN	Binário(a)
CARC	Claim Adjustment Reason Codes
CBHPM	Classificação Brasileira Hierarquizada de Procedimentos Médicos
CFS	Correlation Feature Subset
CID	Classificação Internacional de Doenças
CSE	Consistency Subset Eval
DSS	Decision Support Systems
DT	Decision Tree
EFT	Extremely Randomized Tree
EHR	Electronic Health Records
EUA	Estados Unidos da América
FN	Falso Negativo
FP	Falso Positivo
GMM	Gaussian Mixture Model
GR	Gain Ratio
GS	Genetic Search
GSW	Greedy Stepwise

ICA	Independent Component Analysis
ID3	Iterative Dichotomiser 3
IG	Information Gain
K	Índice Kappa
KNN	K-Nearest Neighbors
KPCA	Kernel PCA
LR	Logistic Regression
LSA	Latent Semantic Analysis
MNB	Multinominal Naive Bayes
NB	Naïve Bayes
NN	Neural Networks
OPS	Operadora de Plano de Saúde
PCA	Principal Component Analysis
PGD	Programação Genética Difusa
PIB	Produto Interno Bruto
PRC	Precision-Recall Curve
R	Recall
RB	Representação Binária
RF	Random Forest
RS	Ranker Search
RT	Random Tree
SUS	Sistema Único de Saúde
SVM	Support Vector Machine
TF	Term Frequency
TF-IDF	Term Frequency Inverse Document Frequency
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

Sumário

1 Introdução

O sistema de saúde brasileiro compreende o Sistema Único de Saúde (SUS) (BRASIL, 2018), responsável pelo atendimento de qualquer cidadão sem contrapartida financeira (MEDEIROS, 2012), e o sistema de saúde suplementar de caráter privado, oferecido pelas Operadoras de Planos de Saúde (OPS). Essas OPSs estão sob regulamentação da Agência Nacional de Saúde Suplementar (ANS) (ANS, 2018a) criada em 2001, com o objetivo de promover, regulamentar e contribuir para o desenvolvimento das ações de saúde no Brasil (SALVATORI; VENTURA, 2012). Nesse sistema híbrido de saúde, o SUS atua como principal e as OPSs de maneira complementar, como é esclarecido pelas Leis nº 9961/98 (BRASIL, 1998) e nº 9961/2000 (BRASIL, 2000). Porém, como descrito por Carvalho, Fortes e Garrafa (2013), na prática, as OPSs oferecem os mesmos serviços do SUS. O problema nesse caso é que a maioria das OPSs são de pequeno porte e não tem recursos suficientes para bancar todo e qualquer requisição realizada por um beneficiário. Assim sendo, muitas dessas OPSs acabam fechando por questões financeiras e pressionando ainda mais o mercado de planos de saúde e o próprio SUS.

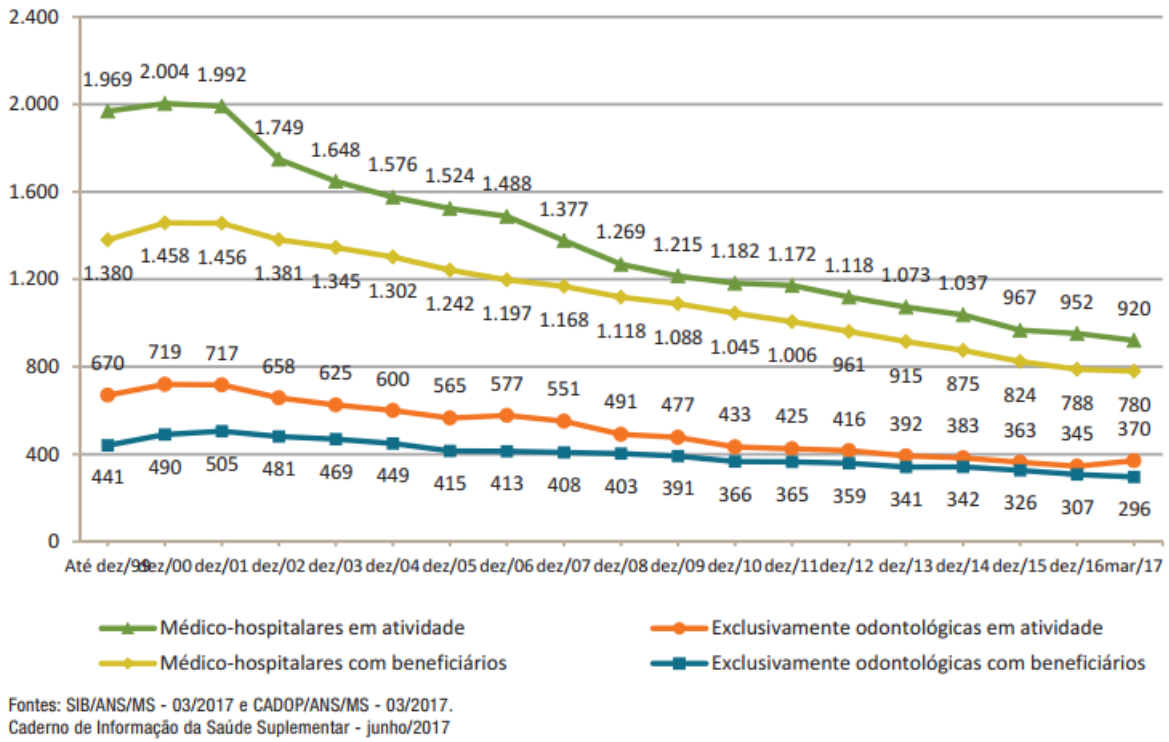

Figura 1 – Operadoras de Planos de Saúde em atividade (Brasil - dezembro/1999-março/2017). Fonte: (ANS, 2018b).

A Figura 1 apresenta um gráfico que acompanha o declínio da atividade das OPSs ao longo dos anos. Nesse gráfico são apresentados informações sobre as operadoras em

atividade e com beneficiário dos ramos médico-hospitalar e odontológico. Segundo o ANS (2018b), desde que a ANS foi criada no início dos anos 2000, até o primeiro trimestre de 2017, o número de operadoras em atividade caiu de 2004 para 920, sendo que apenas 780 das 920 contavam com beneficiários. O mesmo pode ser observado com operadoras de planos odontológicos. Se continuar seguindo essa tendência, a crise no setor pode se agravar ainda mais nos próximos anos. O impacto da insolvência dessas operadoras remanescentes é bastante significativo. Primeiro porque, de acordo com Barros e Beiruth (2016), o mercado de Saúde Suplementar movimenta aproximadamente 5,8% do PIB (Produto Interno Bruto) nacional e atende cerca 25,1% dos brasileiros. Nesse contexto, uma crise neste setor causaria um grande impacto econômico e uma sobrecarga no atendimento já debilitado do SUS. Por isso, estratégias para diminuição dos custos das OPSs são de grande importância.

Uma das principais estratégias utilizadas pelas OPSs para redução de custos é a Regulação Médica. Essa estratégia consiste em uma revisão técnica dos procedimentos/exames/tratamentos para determinar a melhor opção para o beneficiário, evitando requisições desnecessárias. A Figura 2 apresenta um diagrama demonstrando como a regulação acontece. Inicialmente, um profissional de saúde ou uma clínica solicita um exame/procedimento/tratamento a um paciente que então repassa essa solicitação à OPS. Essa solicitação ou requisição passa então por análise de viabilidade conhecida usualmente de avaliação administrativa, ou simplesmente controle, que consiste em uma avaliação da elegibilidade do beneficiário, ao item solicitado, baseado em questões administrativas (carência, adimplência, cobertura do plano de saúde e outros parâmetros não técnicos). Uma vez aprovado administrativamente, é analisado a pertinência técnica da solicitação. Nessa análise, um profissional vinculado à OPS realiza uma avaliação técnica confrontando os protocolos em saúde e a descrição do caso. Dependendo do parecer desse profissional, é dada uma avaliação positiva ou negativa a solicitação. Essa fase que envolve conhecimento técnico especializado é o que chamamos propriamente de regulação.

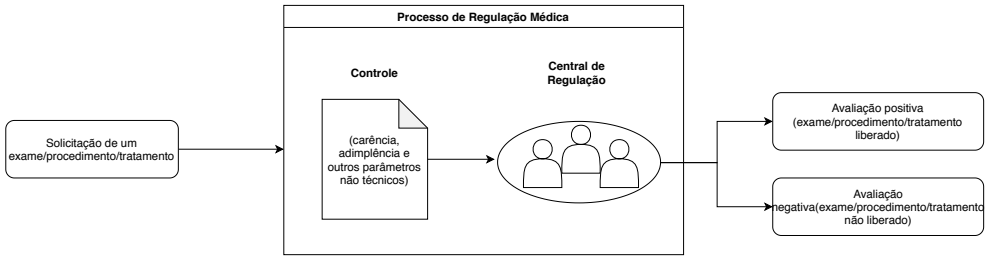

Figura 2 – Funcionamento do processo de regulação de uma OPS.

A regulação por si só é uma forma de controlar gastos, pois ela permite que somente solicitações realmente necessárias sejam aprovadas e isso gera uma economia às OPSs. Porém, a regulação também gera custos, pois é necessário que exista uma central de regulação composta por profissionais disponíveis para análise das solicitações (ARAUJO, 2014). Uma forma de melhorar esse processo, de maneira a reduzir custos, é a utilização

de técnicas de inteligência artificial (IA) como a aprendizagem de máquina (AM).

1.1 Regulação Automatizada

Em um processo de regulação automatizada, é gerado um modelo de aprendizado, baseado nas informações de requisições que ficam armazenadas em uma base de dados. A dinâmica é a mesma da regulação comum, porém há um acréscimo de um sistema inteligente capaz de intermediar o processo entre o Controle e a Central de Regulação. Essa modificação permite que as OPSs reduzam custos administrativos e burocráticos para as solicitações.

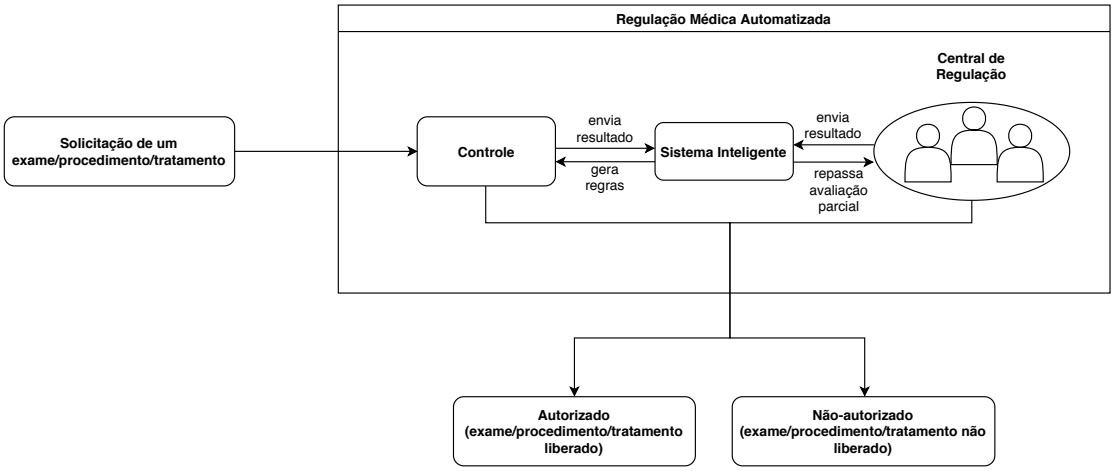

Figura 3 – Funcionamento do processo de regulação médica automatizada.

A Figura 3 apresenta um fluxograma com as etapas do processo de regulação automatizado. Comparando-a com a Figura 2 é possível perceber a existência de um sistema inteligente que funciona como um módulo auxiliar no processo de regulação. Servindo tanto para gerar/atualizar as regras que estão inseridas no Controle, como também para auxiliar na tomada de decisão realizada pela Central de Regulação com o envio de uma avaliação parcial. É importante ressaltar que esse sistema inteligente nada mais é que um sistema de classificação baseado em aprendizado de máquina que aprende as regras de negócio da regulação por meio de treinamento com os resultados enviados tanto pelo Controle quanto pela Central de Regulação.

As vantagens dessa abordagem são, a redução de custos com as etapas de avaliação administrativa, como foi mencionado anteriormente. Além da rapidez na análise das solicitações, proporcionada pelo sistema inteligente. Apesar de o sistema inteligente proporcionar redução de custos e eficiência à regulação, a qualidade desse sistema depende do modelo gerado pela classificação. Dessa forma, a aplicação de técnicas de aperfeiçoamento e modelagem dos dados são de extrema importância para reduzir erros de classificação e consequentemente melhorar a performance do sistema.

1.2 Histórico de Solicitações

O uso de informações de histórico para auxílio na aprendizagem de máquina é um tópico de constante pesquisas científicas devido ao potencial que essas informações trazem ao modelo de aprendizagem. Diversas abordagens, em diferentes áreas de pesquisa, utilizam o histórico de informações para esse fim. Como é o caso de Hu, Zhang e Zhou (2016), que utiliza dados de histórico de fazendas de energia eólica para predizer informações sobre a velocidade do vento em outras fazendas. O histórico é também muito utilizado em pesquisas que envolvem análises de censo (Richards et al., 2014), predição de tempo de espera (KIANPISHEH SAEED JALILI, 2012) (FAN et al., 2018), auxílio no diagnóstico de doenças (ALAM et al., 2019)(KHAMIS; CHERUIYOT; KIMANI, 2014), classificação de risco de crédito ao consumidor (KHANDANI; KIM; LO, 2010), classificação de e-mail como spams (MENAHEM; PUSIZ; ELOVICI, 2012) e etc. Desta forma, pensando no problema de regulação, o histórico poderia trazer benefícios à aprendizagem do modelo de classificação.

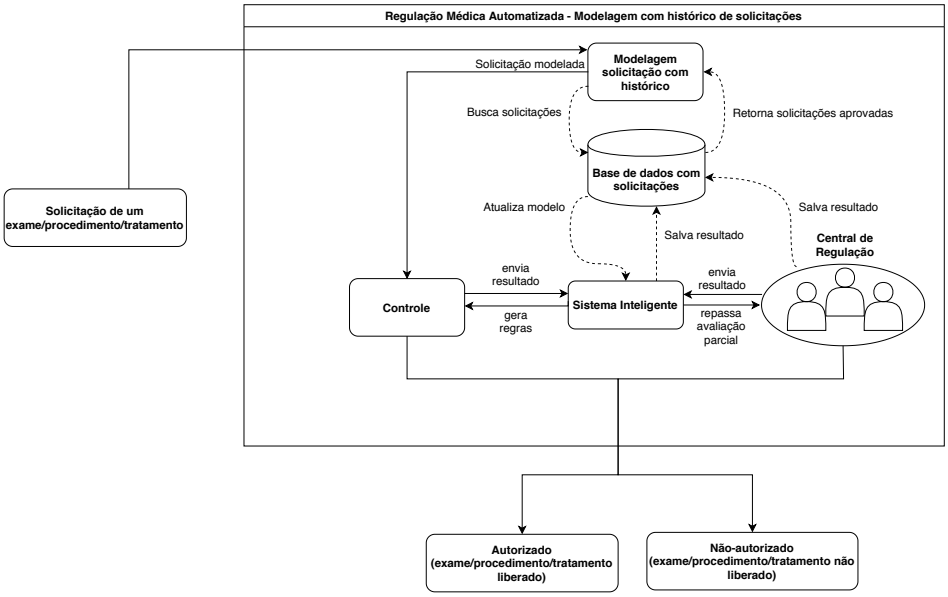

Figura 4 – Funcionamento do processo de Regulação Automatizada com uso de histórico.

No contexto da saúde suplementar, o histórico é representado pelos procedimentos realizados pelo beneficiário, analisados de acordo com um período de tempo especificado ou toda a trajetória desde o primeiro procedimento registrado. No processo de regulação automatizada, essas informações servem de características adicionais ao modelo tradicional, possibilitando um aumento na qualidade das decisões tomadas e, consequentemente, uma redução dos custos relacionados à aprovação ou não de uma requisição. Assim sendo, a informação do histórico entraria no modelo como um módulo de auxílio na avaliação das solicitações, como demonstrado na Figura 4.

O processo de regulação automatizada continua com a mesma dinâmica, porém há

o acréscimo de um novo módulo (Modelagem da solicitação com histórico). A solicitação não é repassada diretamente para o Controle. Primeiro é realizado uma modelagem nova da solicitação onde são acrescidas informações das solicitações passadas do beneficiário. Para criação desse histórico são considerados todos os procedimentos/exames/tratamentos realizados pelo beneficiário até a data solicitação corrente. Essas informações são obtidas da Base de Dados das Solicitações, onde são armazenados todas as informações de todo processo de Regulação. Depois de modelada, a solicitação modificada é repassada ao Controle e os processos da regulação automatizada ocorrem como esperado.

A Figura 5 apresenta uma visão um pouco mais detalhada de como o histórico de solicitações é modelado nesta abordagem. Uma solicitação, como apresentada na figura, apresenta informações comuns de uma solicitação: ID da solicitação, código do beneficiário, CBHPM (Classificação Brasileira Hierarquizada de Procedimentos Médicos), valor, data do procedimento, código da internação, CID (Código Internacional de Doenças). Quando uma solicitação é iniciada no processo de regulação, são buscadas, na base de dados, solicitações passadas do beneficiário. Na Figura 5, são apresentadas todas as solicitações anteriores do beneficiário (Aprovadas e Não-aprovadas). Para inserir essas informações na solicitação atual, são utilizados os códigos CBHPM, pois cada código representa um possível procedimento/tratamento/exame que um beneficiário pode solicitar à OPS. Cada CBHPM é considerado como um novo atributo que apresenta a informação da quantidade de vezes que aquele procedimento/tratamento/exame foi realizado pelo beneficiário. Na figura é possível perceber que o beneficiário realiza a solicitação de uma Tomografia. Ao buscar na base os procedimentos, é possível perceber que duas tomografias foram realizadas no passado, além de uma Anestesia e Internação. As informações dos procedimentos não-realizados (Biópsia e Endoscopia) também são utilizados, pois também é interessante para o aprendizado em regulação o que não é comum no histórico do paciente.

Solicitação de um exame
ID Solicitação
Código Beneficiário
CBHPM
Valor
Código de Internação
Data do Procedimento
...
CID
543987
654321
TOMOGRAFIA
55.68
78320
12/05/2019
K29.7
Busca solicitações beneficiário
Base com solicitações
ID Solicitação
Código Beneficiário
CBHPM
Valor
Código de Internação
Data do Procedimento
...
CID
Autorizado
123456
654321
TOMOGRAFIA
55.68
79650
24/11/2016
K29.7
S
123457
654321
ANESTESIA
38.51
79706
03/07/2016
O34.2
S
123486
654321
INTERNACAO
4.54
79877
08/04/2016
A09
S
221456
654321
ENDOSCOPIA
38.51
80012
06/10/2016
K29
N
422456
654321
BIOPSIA
23.56
80571
01/19/2017
R07.3
N
512356
654321
TOMOGRAFIA
55.68
90123
01/19/2017
K29.7
S
Retorna Solicitações aprovadas
Modelagem da Solicitação com Histórico
ID Solicitação
Código Beneficiário
CBHPM
Valor
Código de Internação
Data do Procedimento
CID
TOMOGRAFIA
ANESTESIA
INTERNACAO
ENDOSCOPIA
BIOPSIA
543987
654321
TOMOGRAFIA
55.68
95010
12/05/2019
K29.7
2
1
1
0
0

1.3 Definição do Problema

Como foi discutido anteriormente, o processo de regulação é uma das principais estratégias utilizadas pelas OPSs para eleger quais solicitações serão realizados ou não. Consequentemente esse processo está diretamente relacionado a dois tipos de custos operacionais primordiais. O primeiro é o custo relacionado à realização de um procedimento, e o segundo é o custo relacionado aos profissionais que ficam a disposição para análise das solicitações. A partir desse problema foram propostas soluções computacionais, como a regulação automatizada, para auxiliar no processo de tomada de decisão e, consequentemente, reduzir custos. Considerando os potenciais benefícios oferecidos pelo histórico de pacientes, foi definida a seguinte questão deu origem a este trabalho: A adição da informação de histórico de beneficiários auxilia o processo de regulação em uma OPS?

Para responder essa questão é necessário realizar outro questionamento: Como inserir o histórico de beneficiários ao processo de Regulação? A resposta é o resultado do estudo apresentado neste trabalho, que apresenta uma metodologia baseada em pré-processamento com modelagem do histórico para aprendizagem da regulação médica.

Definidas as técnicas, já é possível responder ao primeiro questionamento baseado nas seguintes hipóteses:

- Hipótese nula: não há diferença no aprendizado de máquina durante o processo de regulação, quando se usa informações do histórico de atendimentos de um beneficiário.
- Hipótese alternativa: existe diferença no aprendizado de máquina durante o processo de regulação, quando se usa informações do histórico de atendimentos de um beneficiário.

1.4 Objetivos

Nesta seção são apresentados os objetivos geral e específicos necessários para responder aos questionamentos e hipóteses definidas pela pesquisa.

1.4.1 Objetivo Geral

O objetivo principal deste trabalho é melhorar o aprendizado automático do processo de regulação médica, por meio do uso do histórico de beneficiários. Com essa abordagem espera-se aumentar a abrangência da regulação, reduzindo assim custos operacionais e aumentando a agilidade na tomada de decisões. Para cumprir com esse objetivo, serão

aplicadas diferentes técnicas de seleção, transformação e representação de dados visando adaptar as informações do histórico do beneficiário à solicitação realizada.

1.4.2 Objetivos Específico

Durante a pesquisa foram determinados os seguintes objetivos:

- Revisão de literatura sobre técnicas de seleção, transformação e representação de dados, buscando diferentes formas de estruturação a serem aplicadas às informações de histórico de beneficiários;
- Aplicação das principais técnicas de seleção, transformação e representação de dados nas informações do histórico, visando obter diferentes modelos de dados para teste;
- Execução de experimentos, que consistem na aplicação de algoritmos de classificação de forma que os resultados possam ser comparados entre si, evidenciando quais modelagens influenciaram positivamente nos resultados;
- Realização de customização dos modelos estudados de maneira a melhorar os resultados obtidos.

1.5 Visão Geral da Proposta

A fim de cumprir os objetivos propostos, foi desenvolvida uma metodologia composta pelas etapas de Coleta de Dados de Solicitação e Histórico de Solicitações, Pré-processamento, Predição, Avaliação da Performance e apresentação do Modelo de Classificação.

A etapa de coleta dos dados apresenta informações sobre os dados obtidos para realização da pesquisa. Esses dados são analisados separadamente como Dados de solicitação, que são o conjunto de informações que uma solicitação comum apresenta ao processo de regulação, e os dados de histórico, que são a representação de todos os procedimentos/exames/tratamentos realizados pelo beneficiário até o momento da solicitação. A etapa de pré-processamento apresenta todas as transformações que são realizadas na base para prepará-la para a etapa de predição. É dividida em três subetapas: modelagem do histórico, preparação dos dados e seleção/transformação de atributos. A primeira subetapa apresenta com detalhes como o histórico foi modelado, a segunda apresenta diferentes técnicas de representação dos dados além da preparação da base, e a terceira apresenta as principais técnicas de seleção e transformação de atributos para reduzir a dimensionalidade dos dados e aumentar o poder de predição dos classificadores da etapa seguinte. Na etapa de predição, os dados pré-processados servem de entrada para os classificadores: *Random Forest*(RF), *K-nearest neighbors* (KNN), *Support Vector Machines* (SVM) e *Naive Bayes* (NB). A etapa

de avaliação de performance realiza a aplicação de métricas de avaliação de classificadores nos resultados apresentados pela etapa de predição. E por fim, é apresentado um Modelo de Classificação para Regulação Automatizada baseado no histórico dos beneficiários. A Figura 6 apresenta uma versão resumida de todas as etapas da metodologia proposta.

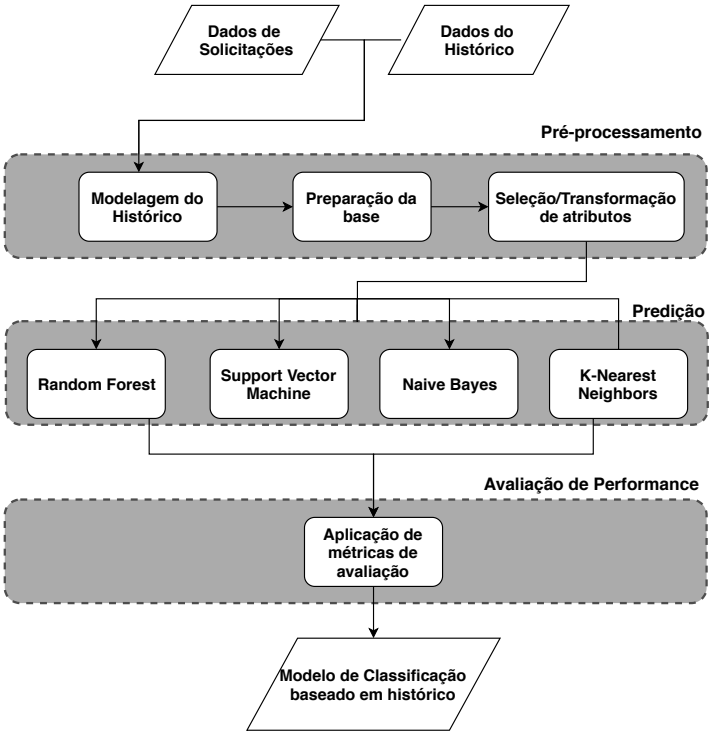

Figura 6 – Fluxograma com a visão geral da proposta.

1.6 Organização da Dissertação

Este documento está estruturado em 6 capítulos. O Capítulo 1 apresenta uma introdução ao conteúdo abordado. No Capítulo 2 é apresentado os principais trabalhos encontrados na literatura sobre o tema. O Capítulo 3 apresenta as principais ferramentas e técnicas utilizadas, tais como: o uso de algoritmos de seleção, extração e construção de atributos, aliados ao uso de classificadores e as métricas de avaliação de resultados. O Capítulo 4 apresenta detalhadamente o sistema proposto e toda abordagem utilizada para o desenvolvimento do trabalho. O Capítulo 5 apresenta os resultados obtidos. Por fim, o Capítulo 6 apresenta as conclusões e trabalhos futuros.

2 Trabalhos Relacionados

Neste capítulo são apresentados os principais trabalhos relacionados à Aprendizagem de Máquina para auxílio em sistemas de tomada de decisão (Decision Support Systems - DSS) com foco na Regulação Médica. Além disso, também são apresentados alguns trabalhos que não são diretamente relacionados ao tema de pesquisa, mas, de alguma forma, contribuíram para o desenvolvimento da metodologia proposta.

2.1 Trabalhos de influência

Até o presente momento não foram encontrados na literatura trabalhos que tratem especificamente do uso do histórico de beneficiários no contexto da Regulação Médica em OPSs, o que torna esta pesquisa um avanço para o estado da arte sob o ponto de vista da Regulação Automatizada.

Tomando como base o contexto da Regulação Médica, alguns trabalhos serviram de influência para a realização deste, como o apresentado por Silva (2011), que apresenta uma proposta de aplicação de Mineração de Dados em uma base de dados da Unimed Catalão, com o objetivo de identificar beneficiários hipertensos ou com suspeitas de presença da doença. Para isso, é feito um pré-processamento nos dados de maneira a obter informações que realmente importassem ao interesse da pesquisa. Ao final do processo, foram selecionadas 331 tuplas, com uma base de treino e outra de teste, onde os atributos selecionados foram separados baseados nos exames comuns pedidos aos pacientes com suspeita de hipertensão, sendo eles: Mapa, Eletrocardiograma, Creatina, Glicose, Uréia e Potássio. Além disso foram considerados como atributos também o Sexo e a Idade do beneficiário. Após a preparação dos dados para teste, foi utilizado o algoritmo ID3 (Iterative Dichotomiser 3) (QUINLAN, 1986) baseado em árvore de decisão, a fim de obter um conjunto de regras que auxiliassem no diagnóstico da doença. Ao final, foram obtidos resultados satisfatórios, com uma taxa de acurácia de 99,09%.

O Trabalho de Marins et al. (2012) apresenta uma abordagem de aprendizagem de máquina em dados de beneficiários de plano de saúde com objetivo de se obter um modelo de aprendizado para predizer a recusa ou não de um procedimento/tratamento/teste. Os dados analisados são de uma OPS brasileira, dessa forma, o processo de atendimento de uma solicitação por parte da OPS segue as normas previstas pela ANS. A base testada é desbalanceada, com mais registros da classe NÃO (Não recusar solicitação) que da de SIM (Recusar solicitação). Foram utilizados os algoritmos PGD (Programação Genética Difusa) e C5.0 para extrair padrões que pudessem auxiliar o processo de tomada de decisão. São realizados dois métodos de treinamento, método *holdout*, onde 2/3 dos dados foram

utilizados para treinamento e 1/3 para teste, e validação cruzada em seis partes. Os resultados foram comparados considerando apenas a taxa de acerto e a avaliação de alguns atributos específicos, considerados atributos metas pelo especialista de domínio. Os autores buscaram criar um conjunto de regras de produção que sintetizem o comportamento da base, quantos e quais atributos tem maior influência no modelo final. Nos testes realizados o PGD obteve melhores resultados de acurácia para ambas as classes do problema, enquanto o C5.0 teve dificuldades de classificar registros da classe SIM. Em relação ao teste com validação cruzada o C5.0 apresentou a melhor taxa de acerto em dois dos três atributos metas, porém o PGD apresentou a maior cobertura para esses atributos.

Outro trabalho de grande destaque, é o apresentado por Araújo, Santana e Neto (2016). Nele, os autores propõem uma metodologia baseada em pré-processamento combinada com algoritmos de Aprendizagem de Máquina para aprender o processo de Regulação. A base de dados utilizada foi uma sobre dados odontológicos, mas a abordagem é criada de modo a englobar dados de qualquer tipo de OPS. Durante a etapa de pré-processamento, a base de dados passa por fase de seleção de atributos manual e automatizada por meio da técnica de ganho de informação, além disso alguns atributos são construídos pelo especialista de domínio. Em seguida são tratados os valores desconhecidos da base e a base final, após o pré-processamento, totaliza 27 atributos. Os testes de classificação são realizados com a base desbalanceada e balanceada por *under-sampling*. Na etapa de classificação foram testados os algoritmos Random Tree (RT), Naive Bayes (NB), Support Vector Machine (SVM) e Nearest-Neighbor (NN). A avaliação ocorreu utilizando o índice Kappa dos melhores classificadores que foram combinados em *ensembles*. O resultado dos classificadores individuais, na base de dados replicada (balanceada) e não-replicada (desbalanceada) revelou que, para a base não replicada, NN e o RT tiveram os melhores resultados com 87% de acurácia para ambos os classificadores e índice Kappa de 0.7 (um nível excelente). Para a base replicada os algoritmos NN e RT também tiveram os melhores resultados com 93% de acurácia e 0.86 de Kappa para ambos algoritmos. Nos testes com o *ensemble* foram testados a base replicada e não-replicada com os critérios de combinação (média, votação com peso e votação sem peso). Os resultados apontaram que o critério de votação com peso obteve os melhores resultados para a base replicada e não-replicada, com 96% de acurácia e kappa de 0.94 para a primeira e 94% de acurácia e 0.85 de kappa para a segunda.

Por último, considerando o tema Regulação Médica, tem-se o trabalho de Saripalli, Tirumala e Chimmad (2017) que apresenta uma proposta baseada em aprendizagem de máquina para predizer o risco de rejeição de solicitações médicas feitas às Operadoras de Planos de Saúde. É importante destacar, que a avaliação feita pelos autores é realizada seguindo as normas e protocolos regulamentados pelo governo dos Estados Unidos da América (EUA). Segundo os autores, existe um problema que é chamado de reprocessamento de solicitações rejeitadas. Esse problema ocorre quando uma solicitação, realizada por um

provedor, é rejeitada pelo pagador por um problema de preenchimento da solicitação. Ao negar, o pagador informa os problemas, o provedor realiza as modificações e a solicitação é reprocessada. Esse reprocessamento tem um grande impacto financeiro, principalmente para as provedoras. Segundo os autores entre 1,38% a 5% das solicitações são negadas pelos planos na primeira submissão. Convertendo esse valor para uma dado monetário, o custo de reprocessamento por ano é em torno de 2 a 10 milhões de dólares. Os autores fazem duas avaliações: uma considerando os dados sem a adição dos atributos gerados pelo especialista (CARC - *Claim Adjustment Reason Codes*) e a outra com o uso dos atributos CARC. As avaliações dos algoritmos de classificação são feitos considerando as métricas acurácia e precisão. No primeiro teste, sem uso dos atributos CARC, o algoritmo CART apresentou 70% de acurácia e 81% de precisão. NN e SVM obtiveram, ambos, 77% de acurácia e 77% de precisão. No segundo teste, com pré-processamento da base e adição de atributos pelo especialista de domínio (CARC), os autores obtiveram os melhores resultados, onde o SVM obteve os melhor resultado em acurácia e precisão (100% para cada métrica). CART vem em seguida com acurácia de 98% e precisão de 97%. Neural Network não obteve bons resultados (45% acurácia e 0% de precisão).

Todos os trabalhos anteriores possuem em comum o uso de algoritmos de aprendizagem máquina supervisionada para auxiliar na geração de um modelo para o contexto da Regulação Médica de uma OPS. A grande diferença da abordagem aqui proposta em relação à elas, é o fato de considerar, além dos dados da solicitação, o histórico de beneficiários como entrada para a aprendizagem. Além disso, a metodologia proposta utiliza várias combinações de técnicas de seleção e transformação de atributos para os dados de histórico com o objetivo de melhorar o aprendizado.

Além do contexto de Regulação Médica, outros trabalhos serviram de influência devido as técnicas de pré-processamento, representação de dados e análise de informações médicas. Um exemplo é o trabalho realizado por Lucini et al. (2017), cujo objetivo é criar um modelo preditivo para prever futuras hospitalizações e liberações de pacientes do departamento de emergência, baseado nas informações de texto presentes nos registros médicos. Os autores dissertam sobre o problema de superpopulação do departamento de emergência, com foco na demanda por leitos hospitalares. São apresentados dados relacionados aos hospitais brasileiros, esclarecendo que o uso de um sistema organizado de alocação e liberação de pacientes que recebem alta fornece um grande benefício para a saúde dos pacientes. Apesar do escopo da pesquisa de Lucini et al. (2017) ser diferente da apresentada neste trabalho, a metodologia proposta e a forma como os autores modelaram as informações serviram de inspiração para a forma como o histórico foi modelado neste trabalho.

Um outro trabalho, que foge ao contexto da Regulação, mas que trouxe um conhecimento agregado que auxilou no desenvolvimento da metodologia proposta foi o realizado

por Miotto et al. (2016). Nele, os autores apresentam uma modelagem de representação das informações dos registro eletrônicos de saúde dos pacientes (*Electronic Health Records* - EHRs) de maneira a melhorar a tomada de decisões clínicas. Para isso, utilizam uma abordagem não-supervisionada de aprendizagem profunda de características baseada em *autoencoders*. Essa abordagem busca capturar informações regulares e dependências agregadas nas EHRs de 700000 pacientes de uma base de dados hospitalar. Os autores dão o nome de *deep patient* ao método proposto, que realiza a avaliação de 76214 pacientes, incluindo 78 doenças de diversos domínios clínicos em diferentes janelas temporais. Os autores compararam a abordagem proposta com outros algoritmos de aprendizagem baseada em características, como o PCA, GMM (*Gaussian Mixture Model*), K-Means e o ICA. Para predizer a probabilidade dos pacientes poderem desenvolver um certo tipo de doença, dado o status clínico corrente. Nesse trabalho há também uma representação de histórico em formato de janelas temporais que podem ser definidas dinamicamente. Na abordagem proposta neste trabalho o histórico não possui esse dinamismo, o período analisado é de cerca de 1 ano e não há separação por intervalos.

Alguns trabalhos secundários, com foco em análises médicas e diagnóstico de doenças também foram consideradas, porém com menos influência na definição da metodologia desenvolvida, como é o caso de Kharya (2012). Nele, os autores discorrem sobre diversas técnicas de Mineração de Dados aplicadas ao diagnóstico de câncer de mama. Em Chimieski e Fagundes (2013) são comparadas as eficiências de algoritmos de Aprendizagem de Máquina baseados em árvores de decisão e redes bayesianas em bases de dados de câncer de mama, doenças dermatológicas e de coluna vertebral. Como resultados, os autores obtiveram melhor classificação com algoritmo *BayesNet* paras as bases de câncer de mama e doenças dermatológicas. Para a base de coluna vertebral, os algoritmos baseados em árvore de decisão se saíram melhor. O trabalho de Antonelli et al. (2013) utiliza informações de histórico de pacientes para diagnóstico de diabetes, sendo uma abordagem baseada em agrupamento utilizando o algoritmo DBSCAN (ESTER et al., 1996). A análise de resultados apresentou grupos com bom grau de separação, mostrando a correlação de cada grupo com os padrões de diagnóstico da diabetes, por fim, os autores comentam o potencial de utilização dessa abordagem para outros diagnósticos de doenças.

2.2 Considerações finais

Neste capítulo foram apresentados estudos que abordam o uso de DSS para tomada de decisão em casos relacionados à ambientes médico-hospitalares e regulação médica. Todos os trabalhos apresentam uma abordagem de pré-processamento aliada à aprendizagem de máquina. Isso demonstra a importância que a preparação de dados traz ao processo de aprendizagem e como isso influencia o resultado final. Neste trabalho, são utilizadas algumas das principais técnicas de pré-processamento baseada em seleção e transformação

de atributos para auxiliar o processo de aprendizagem em regulação com uso de histórico de solicitações.

3 Referencial Teórico

Neste capítulo são apresentadas as técnicas utilizadas neste estudo para permitir o uso de informação de histórico de beneficiários ao processo de Regulação Automatizada de uma OPS. No decorrer do capítulo são detalhadas as principais técnicas relacionadas à representação dos dados, seleção e transformação de atributos. Além disso, são apresentadas técnicas de aprendizagem máquina e métricas de avaliação utilizadas para validação dos resultados.

3.1 Representação dos dados

3.1.1 Representação Binária - RB

A representação binária é uma das formas mais simples de representar informações. Para uma matriz de dados A (m x n), têm-se um conjunto de elementos, onde as colunas representam os atributos da base e as linhas cada um dos registros. Na representação binária, cada elemento $a_{ij}(i \leq m)$ e $(j \leq n)$, é preenchido com 1 (quando há ocorrência do atributo em questão) e 0 (quando não ocorrência do atributo) (GUPTA; LEHAL et al., 2009).

3.1.2 Representação Term-frequency - TF

Essa representação é comumente utilizada para contabilizar a frequência que um determinado termo acontece em um documento. Nela, um termo corresponde a uma atributo do problema (uma coluna da matriz A), enquanto um documento é um registro, ou seja uma linha da matriz A. Como os documentos costumam ter diferentes tamanhos, é possível que um termo apareça mais vezes em documentos mais longos do que nos curtos. Por isso a frequência do termo é dividida pelo comprimento do documento (número total de termos no documento), como uma forma de normalização (SALTON; BUCKLEY, 1988):

$$TF(t,d) = \frac{f(t,d)}{max\{f(t,d) : t \in d\}} \tag{3.1}$$

A Equação 3.1 apresenta a fórmula para o cálculo da TF para um termo (t) presente em um documento (d), onde $f(t,d)$ representa o número de ocorrências do termo (t) no documento(d).

3.1.3 Representação Term-frequency inverse-document frequency - TF-IDF

Essa medida representa uma ponderação usada para avaliar o quão importante um termo é para um documento, considerando a coleção total de documentos. A importância aumenta proporcionalmente ao número de vezes que um termo aparece no documento, porém esse valor é contrabalanceado pela frequência do termo em todos os documentos.

O cálculo do TF-IDF é dividido em duas partes, a primeira é apresentada pelo cálculo $TF(t, d)$ apresentado na Equação 3.1. E o cálculo do $IDF(t, d, D)$ apresentado pela Equação 3.2, que calcula o quão importante um termo t é para o conjunto total de documentos (D). No cálculo TF, todos os termos são considerados igualmente importantes, o que pode ser um problema em alguns casos quando termos como "de"e "que"aparecem com muita frequência mas apresentam pouca importância. O cálculo IDF garante que os termos menos importantes ganhem peso menor que os termos mais raros (SALTON; BUCKLEY, 1988).

$$IDF(t, d, D) = log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right) \tag{3.2}$$

Para se obter a ponderação de um termo é necessário realizar o produto de $TF(t, d)$ e $IDF(t, d, D)$. Assim sendo, o cálculo TF-IDF é apresentado pela Equação 3.3:

$$TF\_IDF(t, d, D) = TF(t, d) * IDF(t, d, D) \tag{3.3}$$

3.2 Seleção de Atributos

A seleção de atributos é uma técnica que consiste em escolher um conjunto M de atributos de um conjunto original N, onde $M \leq N$. Em geral, a escolha do melhor subconjunto de atributos é caracterizado como um problema de busca de acordo com algum critério de avaliação. Com o uso dessa técnica é possível reduzir a dimensionalidade dos dados, aumentar a performance dos algoritmos de aprendizagem, melhorar a acurácia de algoritmos de classificação e a compreensão dos dados.

O processo geral de seleção de atributos possui quatro etapas bem definidas mostrada na Figura 7 (KUMAR; MINZ, 2014).

- Geração de subconjunto de atributos
- Avaliação de subconjuntos
- Critério de parada
- Validação do resultado

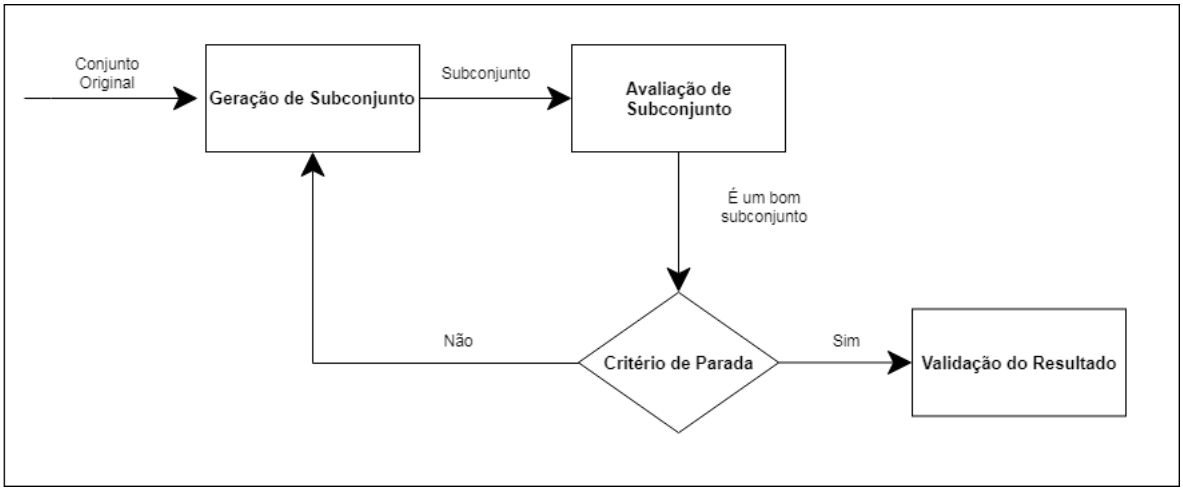

Figura 7 – As quatro etapas fundamentais no processo de seleção de atributos. Fonte: (KUMAR; MINZ, 2014)

Inicialmente o Conjunto Original (Figura 8) de atributos passa primeiramente pela etapa de Geração de Subconjunto. Nessa etapa são aplicados algoritmos de otimização baseados em heurísticas de busca com o objetivo de gerar um subconjunto candidato para a próxima etapa. As estratégias de geração de subconjunto conhecidas são: *sequential forward selection* (Figura 9), *sequential backward selection* (Figura 10), *randomly generation* (Figura 11) e *exhaustively generation* (Figura 12) (MOTODA; LIU, 2002). Na primeira o subconjunto é gerado sequencialmente, partindo de um conjunto vazio e gradualmente é adicionado um atributo por vez. Já na segunda, que também utiliza estratégia sequencial, inicialmente o conjunto começa com todos os atributos e a cada passo é removido um. Em *randomly generation* o subconjunto é gerado aleatoriamente dentro do espaço possível de possibilidades de subconjuntos (2^N onde N é o número de atributos). E por fim em *exhaustively generation* é gerado todos os subconjuntos dentro do espaço de possibilidades (2^N).

Figura 8 – Exemplo de problema com quatro atributos.

Cada novo subconjunto precisa ser avaliado por algum critério. Na etapa de Avaliação de Subconjunto, os critérios de avaliação de atributos são categorizados em dois grupos, o primeiro é o independente, que explora as características dos dados sem envolver algoritmos de classificação para avaliar o subconjunto de atributos. O segundo é o dependente, que necessita de algoritmos de classificação no processo de seleção de atributos para avaliar um subconjunto. Para terminar um processo de seleção de atributos é necessário que um Critério de Parada seja determinado. Após uma avaliação de subconjunto atender um critério de parada, um resultado é retornado, sendo esta a última etapa.

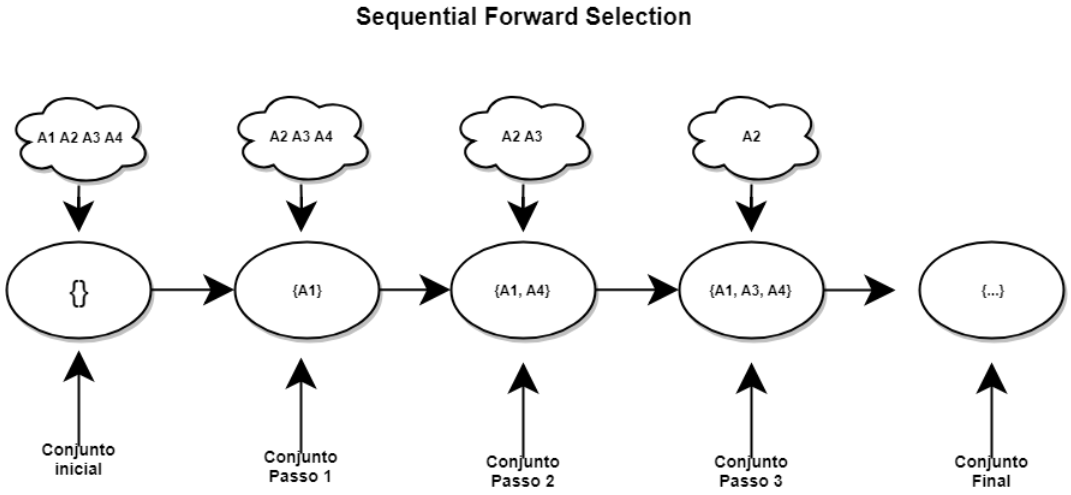

Figura 9 – Exemplo de geração de subconjunto usando *sequential forward selection*.

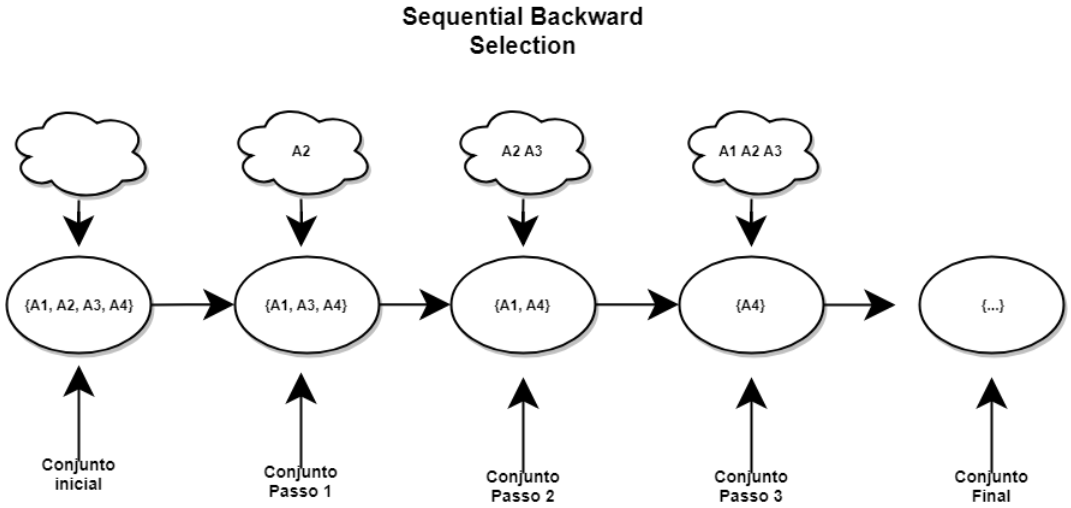

Figura 10 – Exemplo de geração de subconjunto usando *sequential backward selection*.

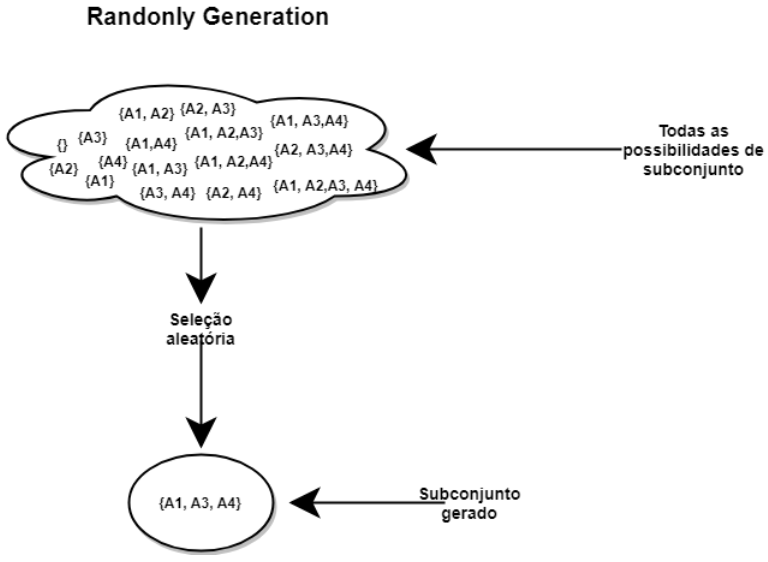

Figura 11 – Exemplo de geração de subconjunto usando *randomly generation*.

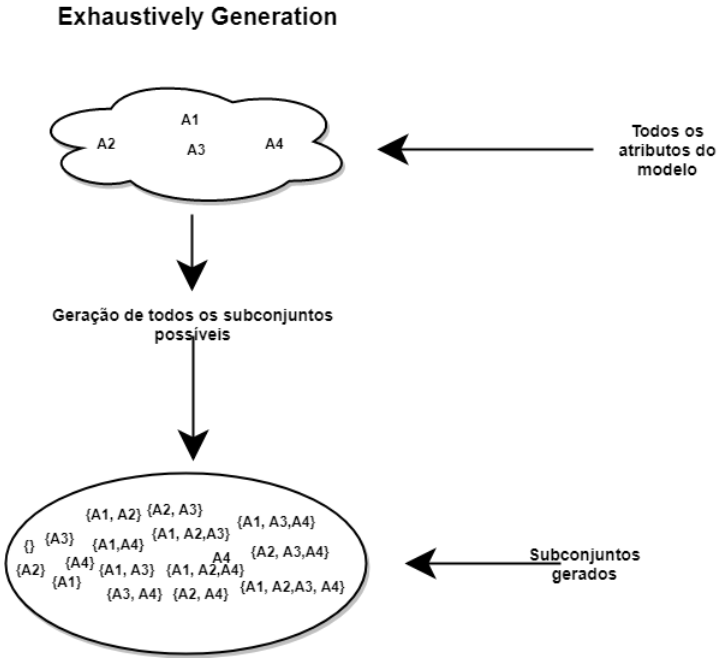

Figura 12 – Exemplo de geração de subconjunto usando *exhaustively generation*.

Segundo Kohavi e John (1997), a seleção de atributos pode ser dividida em três categorias: filtro, *wrapper* e *embedded*. Os filtros selecionam os atributos independentemente sem o auxílio de algoritmos de classificação. Nos *wrappers*, os algoritmos de mineração influenciam diretamente no processo de busca dos subconjuntos de atributos. Os *wrappers* costumam apresentar melhores resultados, porém o custo computacional é elevado em comparação aos filtros. Já os *embedded* selecionam o conjunto de atributos no próprio processo de construção do modelo de classificação durante a fase de treinamento e não são geralmente específicos para um dado algoritmo de classificação. Em outras palavras a seleção de atributos ocorre "embutida"na classificação.

Neste trabalho foram utilizados somente os algoritmos baseados em filtros e *wrappers*. Os algoritmos do tipo *embedded* não foram utilizados porque a etapa de seleção de atributos, na abordagem proposta, se encontra separada da etapa de classificação.

3.2.1 Algoritmos de Geração de Subconjunto

- ***Greedy Stepwise***: É um algoritmo guloso que realiza uma *sequential forward search* ou uma *sequential backward search*. Desta forma pode partir de um subconjunto com todos os atributos ou com nenhum. O critério de parada é quando a adição ou deleção de um atributo resulta em um decréscimo na avaliação (WITTEN et al., 2016).

- ***Best First***: Realiza a busca dentro do espaço de atributos utilizando o algoritmo guloso *hillclimbing* com *backtracking*. Assim como o algoritmo *greedy stepwise*, permite realizar tanto o *forward* quanto o *backward search*, além de também permitir a busca

em ambas as direções. O critério de parada é dado pelo número de iterações em que a busca não retornou uma melhora em relação à solução atual. A quantidade de iterações é um parâmetro do algoritmo e a partir dele também é possível controlar o nível de *backtracking* (WITTEN et al., 2016).

- ***Linear Forward Selection***: É uma extensão do best first porém leva em consideração um número n de atributos pré-determinado e a direção da busca é *forward*. A seleção dos n atributos pode ser de maneira fixa, ou ela pode ser incrementada a cada passo. Inicialmente o algoritmo pode utilizar a ordem inicial dos atributos ou ordená-los utilizando um parâmetro de avaliação. O critério de parada é o mesmo do *best first*(GUTLEIN et al., 2009).

- ***Genetic Search***: Implementa a busca utilizando a versão simples de algoritmo genético proposto por Goldberg e Holland (1988). A geração da população inicial é feita de maneira randômica (*randomly generation*) e a quantidade de subconjuntos gerados, bem como a taxa de *crossover* e mutação são determinados inicialmente antes do algoritmo iniciar a busca. A seleção dos indivíduos para reprodução utiliza a técnica da roleta, onde, aos indivíduos é dada uma fatia correspondente ao seu *fitness*. O critério de parada é dado pela quantidade máxima de gerações determinado inicialmente.

- ***Ranker Search***: Utiliza um avaliador para 'rankear' todos os atributos. Para isso utiliza a técnica *forward selection* para selecionar os melhores atributos um por vez, primeiro o melhor, depois o segundo melhor e assim sucessivamente. No final retorna todos os atributos *rankeados* por ordem de avaliação dada por algoritmos simples como: ganho de informação, razão de ganho e *Relief* (HALL; HOLMES, 2003).

Devido ao tamanho do problema, algoritmos de baseados em busca exaustiva foram desconsiderados para este trabalho.

3.2.2 Algoritmos de Avaliação de Subconjunto

- **Ganho de informação (QUINLAN, 1986)** : É uma técnica de seleção de atributos baseada em filtro, que utiliza da informação de entropia para selecionar os atributos de um subconjunto. Conceitualmente falando, dado um atributo a, cujo domínio é $\{a_1, a_2, ..., a_k\}$ para $k \geq 1$, têm-se a probabilidade p_j, $1 \leq i \leq k$ para cada valor a_i, definida como razão entre o número de instâncias da base em que ocorre o valor a_i para o atributo a e o número total de instâncias. A entropia é dada pela seguinte equação:

$$h(a) = -\sum_{i=1}^{k}[p_i * \log_2(p_i)] \tag{3.4}$$

Além de calcular a entropia dos atributos é necessário calcular a entropia da classe, dada por $h(c)$, que pode ser calculado de maneira similar ao do atributo. Considerando p_j a razão entre o número de instâncias em que o valor c_i da classe, $1 \leq j \leq m$, ocorre na base e o número total de instâncias.

Desta forma, seja a probabilidade $p_{j|i}$ a razão entre o número de instâncias da base que pertencem à classe c_j em que ocorre o valor a_i do atributo a, e o número total de instâncias da base, a entropia condicional $h(c|a)$ é dada por:

$$h(c|a) = -\sum_{i=1}^{k}\sum_{j=1}^{m}[p_{j|i} * \log_2(\frac{p_{j|i}}{p_i})] \tag{3.5}$$

Assim, pode se concluir, que quanto mais informação um atributo a tiver em relação à uma classe c, menor será a entropia condicional. De posse dessas informações já é possível calcular o ganho de informação do atributo a em relação à classe c, definido como a diferença entre a entropia da classe e a entropia condicional da classe dado o atributo a, como demonstrado na equação a seguir:

$$infoGain(c|a) = h(c) - h(c|a). \tag{3.6}$$

A Tabela 1 apresenta um exemplo de aplicação da informação de ganho. Nela são apresentadas informações meteorológicas relacionadas a prática de tênis.

Tabela 1 – Dados representando condições para se jogar ou não tênis baseado em informações das condições meteorológicas. Fonte: (WITTEN et al., 2016)

Instâncias	**Expectativa**	**Temperatura**	**Humidade**	**Vento**	**Jogar Tênis**
1	Sol	Quente	Alta	Fraco	Não
2	Sol	Quente	Alta	Forte	Não
3	Nublado	Quente	Alta	Fraco	Sim
4	Chuva	Moderada	Alta	Fraco	Sim
5	Chuva	Fresco	Normal	Fraco	Sim
6	Chuva	Fresco	Normal	Forte	Não
7	Nublado	Fresco	Norma	Forte	Sim
8	Sol	Moderada	Alta	Fraco	Não
9	Sol	Fresco	Normal	Fraco	Sim
10	Chuva	Moderada	Normal	Fraco	Sim
11	Sol	Moderada	Normal	Forte	Sim
12	Nublado	Moderada	Alta	Forte	Sim
13	Nublado	Quente	Normal	Fraco	Sim
14	Chuva	Moderada	Alta	Forte	Não

Para calcular-se a informação de ganho de um determinado atributo dentre o conjunto de atributos do problema, é necessário calcular a entropia total do conjunto de informações, ou seja, a entropia da classe utilizando a Equação 3.1. De acordo com a Tabela 1 têm-se 5 exemplos positivos e 9 negativos, a entropia total é dada por:

$$h(c) = -(\frac{9}{14}) * log_2(\frac{9}{14}) - (\frac{5}{14}) * log_2(\frac{5}{14}) = 0,939$$

Uma vez calculada a entropia do conjunto é possível aplicar a informação de ganho para selecionar o melhor atributo para ser o primeiro nó de uma árvore de decisão, por exemplo.

Para calcular o ganho de informação para o primeiro atributo do modelo (Expectativa) é necessário realizar o cálculo da frequência relativa de cada valor que o atributo Expectativa pode assumir (Sol, Nublado, Chuva), juntamente com a ocorrências das classes do problema (Sim, Não). Assim sendo, têm-se a seguinte tabela:

Tabela 2 – Tabela com valores para cálculo de informação de ganho para atributo expectativa.

Valores	Frequência relativa	Não	Sim
Sol	5	3	2
Nublado	4	0	4
Chuva	5	2	3

$$infoGain(c|expectativa) = h(c) - (\frac{5}{14} * h(sol)) - (\frac{4}{14} * h(nublado)) - (\frac{5}{14} * h(chuva))$$

$$infoGain(c|expectativa) = 0,939 - 0,347 - 0 - 0,347 = 0,245$$

Fazendo esse processo para os outros atributos, obtêm-se:

$$infoGain(c|humidade) = 0,151$$

$$infoGain(c|vento) = 0,048$$

$$infoGain(c|temperatura) = 0,029$$

A conclusão que se pode chegar é que o atributo Expectativa é o melhor dentre os demais para ser a raiz da árvore que se quer construir.

- **Razão de ganho (QUINLAN, 1986)**: Também uma técnica baseada em filtro, a razão de ganho pondera o valor obtido pelo cálculo de informação de ganho, esse método surgiu devido ao fato de o cálculo da informação de ganho superestimar os atributos que apresentam mais ocorrências. Desta forma o cálculo da razão de ganho é apresentado desta forma:

$$gainRatio = \frac{infoGain(c|a)}{h(a)} \tag{3.7}$$

- ***Correlation-based Feature subset*** *(HALL, 1999):* Uma outra maneira de eliminar atributos redundantes ou irrelevantes é selecionando um subconjunto de atributos que individualmente se correlacionam bem com a classe mas apresentam pouca intercorrelação com outros atributos. A correlação entre um atributo A e B pode ser calculada usando 'incerteza simétrica'(*symmetric uncertainty*), demonstrada na Equação 3.8.

$$u(a|b) = 2 * \frac{h(a) + h(b) - h(a|b)}{h(a) - h(b)} \tag{3.8}$$

  Onde h é o calculo da entropia como demonstrado em 3.2.2. A entropia de $h(a|b)$, é obtida a partir da junção das probabilidades de todas as combinações de A e B. A incerteza simétrica é sempre um valor entre 0 e 1. A *correlation-based feature selection* determina o quão bom um conjunto de atributos é usando:

$$\frac{\sum_j u(a_j|c)}{\sqrt{\sum_i \sum_j u(a_i|a_j)}} \tag{3.9}$$

  Onde c é o atributo classe e os índices i e j percorre todos os atributos no conjunto. Se todos os m atributos no subconjunto correlacionam perfeitamente com a classe e um com o outro, o numerador se torna m e o denominador $\sqrt{m^2}$, que é também m. Consequentemente o resultado final será 1, que é o máximo que a função pode obter (o mínimo é 0). Claramente não é o ideal, pois o que se quer é evitar atributos redundantes.

- ***Consistency subset eval*** *(LIU; SETIONO et al., 1996):* Avalia o valor de um subconjunto de atributos a partir de um critério de inconsistência. O algoritmo funciona da seguinte forma: inicialmente é gerado um subconjunto S aleatório, de um conjunto N com todos os atributos. Se o número de atributos (C) de S é menor que a solução atual ($C < C_{best}$), os dados D com os atributos descritos em S são avaliados de acordo com um critério de inconsistência. Se a taxa de inconsistência é menor ou igual a um valor γ especificado, C_{best} e S_{best} são substituídos pelos novos valores C e S respectivamente; Se $C = C_{best}$ e o critério de inconsistência é satisfeito, então uma solução igualmente boa é selecionada e retornada pelo algoritmo.

  O critério de inconsistência ($IconCheck(S, D) \leq \gamma$) é a chave do algoritmo. Este critério especifica a extensão da redução de dimensionalidade que é aceitável. O valor de inconsistência dos dados descritos pelos atributos são checados de acordo com uma taxa γ previamente especificada. Se o valor retornado for menor que γ, significa que a redução da dimensionalidade é aceitável. O valor de γ é 0 por padrão e o valor de inconsistência é calculado da seguinte forma:

- Duas instâncias são consideradas inconsistentes se diferem apenas no valor atributo classe;
- Para todas as instâncias que possuem valores de atributos iguais (sem considerar o atributo classe), a inconsistência é calculada pelo número de instâncias menos o maior número de uma determinada classe. Exemplo: Há n instâncias semelhantes, entre elas, c_1 instâncias pertencem à classe 1, c_2 à classe 2 e c_3 à 3, onde $c_1 + c_2 + c_3 = n$. Se c_3 é o maior que os três, o valor da inconsistência será $(n - c_3)$;
- A taxa de inconsistência é soma de todas as inconsistências divididas pelo número total de instâncias.

Por exemplo, utilizando a Tabela 1 como a base de dados de entrada para o problema, têm-se:

- Inicialmente $S_{best} = \{Expectativa(E), Temperatura(T), Humidade(H), Vento(V)\}$ e $C_{best} = 4$ e $\gamma = 0$
- Supondo que seja gerado o subconjunto $S = \{E, H\}$ onde $C = 2$. Como $C < C_{best}$, é calculada a inconsistência dos atributos do subconjunto S, onde os dados D são como demonstrados na Tabela 3.
- O critério de inconsistência (IconCheck(S,D)) é calculado da seguinte forma para os dados D: Primeiro são verificadas quais instâncias são inconsistentes (possuem valores iguais para os atributos Expectativa e Humidade, diferindo apenas o valor do atributo classe (Jogar Tênis)). Desta forma as instâncias que estão dentro dessas condições são: Instâncias 4 e 14 = $\{Chuva, Alta\}$, Instâncias 5,6 e 10 = $\{Chuva, Normal\}$. Para calcular a inconsistência para essas instâncias é feito o cálculo $(n - c_i)$, onde n é o número de instâncias que se repetem com os determinados valores de atributo e c_i é a quantidade de instâncias da classe com maior número. Ou seja para as Instâncias 5,6 e 10, o valor de $n = 3$ e a classe com maior número de ocorrências é o Sim (2 ocorrências em 5 e 10), assim o valor da inconsistência é $(3 - 2)$. Para calcular a taxa de inconsistência total é realizado um somatório de todas as inconsistências. Deste modo a taxa de inconsistência é dada por:

$$IconCheck(S, D) = \frac{(n_{1,4} - c_{sim}) + (n_{5,6,10} - c_{sim})}{N}$$

$$IconCheck(S, D) = \frac{(2 - 1) + (3 - 2)}{14} = \frac{2}{14} = 0,14$$

Como o valor de $IconCheck(S, D)$ não é menor ou igual a γ então o subconjunto S não substitui o S_{best} atual.

– Já para um subconjunto $S = \{E, H, V\}$ não há instâncias inconsistentes, ou seja o $InconCheck(S, D) = \frac{0}{14}$. Dessa forma o S_{best} é substituído por $\{E, H, V\}$ e $C = 3$.

– Esse processo é repetido até que um critério de parada seja alcançado.

Tabela 3 – Dados representando condições para se jogar ou não tênis baseado em informações das condições meteorológicas, remoção dos Atributos Vento e Temperatura

Instâncias	**Expectativa**	**Humidade**	**Jogar Tênis**
1	Sol	Alta	Não
2	Sol	Alta	Não
3	Nublado	Alta	Sim
4	Chuva	Alta	Sim
5	Chuva	Normal	Sim
6	Chuva	Normal	Não
7	Nublado	Normal	Sim
8	Sol	Alta	Não
9	Sol	Normal	Sim
10	Chuva	Normal	Sim
11	Sol	Normal	Sim
12	Nublado	Alta	Sim
13	Nublado	Normal	Sim
14	Chuva	Alta	Não

- ***Relief*** *(KONONENKO, 1994):* A ideia do algoritmo *Relief* é estimar a qualidade dos seus atributos baseado em como seus valores distinguem das instâncias próximas a eles. Para cada instância, o algoritmo busca por dois vizinhos mais próximos: um que possui a mesma classe e outro que não possui. Baseado nesses valores é calculado a dependência entre os atributos por meio da análise das instâncias próximas.

  O cálculo por trás do funcionamento deste algoritmo funciona da seguinte forma: é definido inicialmente um vetor de pesos $W|a|$, inicializado com 0 para todos os atributos, um valor t, que corresponde ao número de instâncias de treinamento que devem ser selecionados aleatoriamente da base. Para cada uma das instâncias R, são encontrados instâncias vizinhas que compartilham o mesmo valor de classe (denominado *hit* e representado por H). A mesma busca é feita por instâncias vizinhas cujo os valores de classe não coincidem com as das instâncias R (denominado *miss* e representado por M). Para cada um dos atributos da instância, é ajustado o peso desse atributo de acordo com a soma das diferenças entre a instância selecionada e as instâncias *hit* e *miss*, de acordo com a equação:

$$W|a| = W|a| - \frac{dist(a, R, H)}{t} + \frac{dist(a, R, M)}{t}, \tag{3.10}$$

  A função *dist* é usada para calcular a diferença entre os valores de um atributo para duas instâncias, variando de 0 a 1. Para atributos nominais é definido 1 para valores diferentes e 0 para valores iguais. Quando os atributos são contínuos, a diferença é calculada normalmente, onde o resultado é normalizado para o intervalo [0, 1].

Como resultado, tem-se um vetor de pesos atualizados para todos os atributos da base seguindo a proporção: quanto maior o peso, mais relevante é o atributo.

Um exemplo de como esse algoritmo funciona é apresentado a seguir. Inicialmente é escolhido aleatoriamente um conjunto t de instâncias como apresentado na Tabela 4 e os pesos para cada atributo é inicializado com 0 ($W|E| = 0, W|T| = 0, W|H| = 0, W|V| = 0$). Em seguida é escolhido uma instância R para atualização dos pesos, nesse exemplo foi escolhido a instância 8 ($R = 8$), lembrando que este cálculo é feito para todas as instâncias do conjunto de treino.

Determinada a instância, é calculado o *Hit* e o *Miss* para a instância específica, onde $H = 6$ (instância mais próxima com o mesmo valor da classe) e $M = 4$ (instância mais próxima cujo valor de classe sejam diferentes). Terminada essa etapa os pesos para cada atributo é atualizado, conforme demonstrado a seguir:

$$W|E| = W|E| - \frac{dist(E, 8, 6)}{t} + \frac{dist(E, 8, 4)}{t}$$

$$W|E| = 0 - \frac{1}{4} + \frac{1}{4} = 0$$

Para os outros atributos, têm-se:

$$W|T| = 0 - \frac{1}{4} + \frac{0}{4} = -0,25$$

$$W|H| = 0 - \frac{1}{4} + \frac{0}{4} = -0,25$$

$$W|V| = 0 - \frac{1}{4} + \frac{0}{4} = -0,25$$

Esse processo é feito para todas as instâncias da base de treino.

Tabela 4 – Instâncias selecionadas da Tabela 1 para exemplo do algoritmo *Relief*

Instâncias	**Expectativa**	**Temperatura**	**Humidade**	**Vento**	**Jogar Tênis**
1	Sol	Quente	Alta	Fraco	Não
4	Chuva	Moderada	Alta	Fraco	Sim
6	Chuva	Fresco	Normal	Forte	Não
8	Sol	Moderada	Alta	Fraco	Não

- ***Wrapper subset eval*** *(KOHAVI; JOHN, 1997):* Os *wrappers*, diferentemente dos filtros, dependem de um algoritmo de classificação para auxiliar na avaliação dos subconjuntos de atributos (Figura 13). Normalmente apresentam resultados melhores que os filtros (HALL; HOLMES, 2003), pois a seleção de atributos é guiada por um algoritmo de predição que geralmente é utilizado novamente na classificação. A

grande desvantagem dessa técnica para os filtros é o custo computacional, visto que um algoritmo de predição é executado diversas vezes até um que subconjunto bom o suficiente seja selecionado.

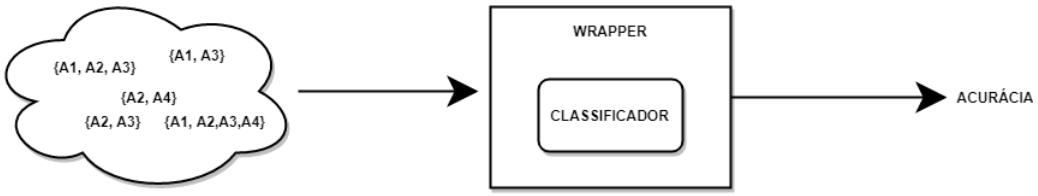

Figura 13 – Exemplo ilustrativo do funcionamento de um algoritmo do tipo *wrapper*. Inicialmente é escolhido um subconjunto de atributos que se deseja avaliar, esse subconjunto serve de entrada para o algoritmo que possui um classificador que utiliza de acurácia preditiva para realizar a avaliação.

3.3 Transformação de Atributos

A transformação de atributos é uma técnica que extrai um novo conjunto de atributos em relação ao conjunto original por meio de uma função de mapeamento. Assumindo que inicialmente há n atributos $A_1, A_2, ..., A_n$, após o processo de extração, tem-se um novo conjunto $B_1, B_2, ..., B_m (m < n), B_i = F(A_1, A_2, ..., A_n)$ onde F_i é a função de mapeamento utilizada para transformar os atributos originais em uma nova representação (MOTODA; LIU, 2002).

Esta técnica pode ser utilizada no contexto de redução de complexidade dos dados, para isso utiliza uma representação do espaço de atributos como uma combinação linear dos atributos de entrada. Apesar de existirem diversas abordagens de transformaçãp/extração de atributos as mais populares são *Principal Component Analysis* (PCA), *Kernel Principal Component Analysis* (KPCA), *Independent Component Analysis* (ICA) e *Latent Component Analysis* (LSA).

3.3.1 Principal Component Analysis - PCA

Em uma base de dados com k atributos numéricos, é possível visualizar os dados como uma 'nuvem' de pontos no espaço $k-$dimensional. Os atributos representam as coordenadas do espaço, mas os eixos utilizados, o sistema de coordenadas em si, é arbitrário (WITTEN et al., 2016).

A ideia do *principal component analysis* é usar um sistema de coordenadas especial para estruturar a 'nuvem' de pontos em uma forma menos complexa, para isso é realizado uma transformação no sistema de coordenadas. Esta transformação acontece da seguinte forma: o primeiro eixo é colocado na direção de maior variância de pontos para maximizar a variância ao longo do eixo. Os outros eixos são colocados sempre perpendicular ao

primeiro, onde cada um é colocado de maneira a maximizar sua partição na variância restante.

Tecnicamente, o PCA pode ser entendido como o cálculo da matriz de covariância dos pontos das coordenadas originais que são diagonalizadas para se obter os autovetores. Estes autovetores nada mais são que os eixos do espaço transformado ordenados pelos autovalores, pois cada autovalor dá a variância ao longo do eixo. Esse cálculo pode ser sintetizado em 5 etapas como demonstradas abaixo:

- **Etapa 1: Preparação dos dados**

  Inicialmente é necessário obter os dados e formatá-los para as próximas etapas. Para exemplificar o funcionamento do PCA será utilizado os dados da Tabela 1. Como os dados referentes a essa Tabela são todos do tipo nominal é necessário adaptar as informações para o modelo numérico. A modelagem para cada atributo foi realizado da seguinte maneira: Expectativa $\{Sol = 3, Nublado = 2, Chuva = 1\}$, Humidade = $\{Alta = 3, Normal = 2\}$, Temperatura = $\{Quente = 3, Moderada = 2, Fresco = 1\}$ e Vento = $\{Fraco = 1, Forte = 2\}$. A Tabela 5 apresenta as informações transformadas para os valores numéricos correspondentes.

Tabela 5 – Dados da Tabela 1 com transformação numérica dos valores dos atributos.

Instâncias	**Expectativa**	**Temperatura**	**Humidade**	**Vento**	**Jogar Tênis**
1	3	3	3	1	Não
2	3	3	3	2	Não
3	2	3	3	1	Sim
4	1	2	3	1	Sim
5	1	1	2	1	Sim
6	1	1	2	2	Não
7	2	1	2	2	Sim
8	3	2	3	1	Não
9	3	1	2	1	Sim
10	1	2	2	1	Sim
11	3	2	2	2	Sim
12	2	2	3	2	Sim
13	2	3	2	1	Sim
14	1	2	3	2	Não

- **Etapa 2: Normalização dos dados a partir da média**

  Para o PCA funcionar perfeitamente é necessário subtrair a média de valores de cada atributo do valor do atributo em cada instância($x - \overline{x}$). Dessa forma foram calculadas as médias para cada atributo: Media_Expectativa = 2, Media_Temperatura = 2, Media_Humidade = 2,5 e Media_Vento = 1,428. A Tabela 6 apresenta os valores normalizados dos dados.

- **Etapa 3: Calcular a matriz de covariância**

Tabela 6 – Tabela com dados normalizados a partir da média.

Instâncias	**Expectativa**	**Temperatura**	**Humidade**	**Vento**	**Jogar Tênis**
1	1	1	0,5	-0,42	Não
2	1	1	0,5	0,58	Não
3	0	1	0,5	-0,42	Sim
4	-1	0	0,5	-0,42	Sim
5	-1	-1	-0,5	-0,42	Sim
6	-1	-1	-0,5	0,58	Não
7	0	-1	-0,5	0,58	Sim
8	1	0	0,5	-0,42	Não
9	1	-1	-0,5	-0,42	Sim
10	-1	0	-0,5	-0,42	Sim
11	1	0	-0,5	0,58	Sim
12	0	0	0,5	0,58	Sim
13	0	1	-0,5	-0,42	Sim
14	-1	0	0,5	0,58	Não

A covariância é calculada de maneira parecida à variância, cujo cálculo é dado por:

$$cov(x,y) = \frac{\sum_{i=1}^{n}(X_i - \overline{X})(Y_i - \overline{Y})}{(n-1)} \tag{3.11}$$

A covariância sempre é calculada entre duas dimensões, mesmo que os dados tenham mais que duas, ou apenas uma (neste caso é calculada a covariância da dimensão com ela mesma). Por exemplo, em dados com 3 dimensões (x, y, z), terão que ser calculados todas as combinações para dois pares de valores 3-dimensional. Para dados n-dimensional, terão de ser calculados $\frac{n!}{(n-2)!*2}$ valores de covariância. Deste modo, o cálculo da matriz de covariância para dados de 3 dimensões, por exemplo, gerará uma matriz de 3 linhas e 3 colunas, tendo como valores:

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

O mesmo pode ser generalizado para dados com outras dimensões.

Para o exemplo em questão os dados possuem 4 dimensões, desta forma a matriz de covariância será 4 x 4. O resultado do cálculo da matriz de covariância para os dados da Tabela 6 é dado pela matriz *cov*:

$$cov = \begin{pmatrix} 0.76923077 & 0.23076923 & 0.07692308 & 0. \\ 0.23076923 & 0.61538462 & 0.23076923 & -0.07692308 \\ 0.07692308 & 0.23076923 & 0.26923077 & 0. \\ 0. & -0.07692308 & 0. & 0.26373626 \end{pmatrix}$$

- **Etapa 4: Cálculo dos autovetores e dos autovalores da matriz de covariância**

Como a matriz de covariância é quadrada, é possível calcular os autovalores e autovetores para esta matriz. Os autovetores determinam as direções do novo espaço de atributos e os autovalores determinam sua magnitude. Em outras palavras, os autovetores apresentam a variância dos dados ao longo dos eixos dos novos atributos. Para a matriz de covariância apresentada anteriormente, foram calculados os autovetores e autovalores apresentados a seguir:

$$autovetores = \begin{pmatrix} -0.72918013 & -0.67743377 & -0.07733456 & -0.0583026 \\ -0.62309911 & 0.6146496 & 0.4835888 & 0.00975897 \\ -0.27523283 & 0.35916489 & -0.81829251 & 0.35446417 \\ 0.06550358 & -0.18517545 & 0.30092958 & 0.93319919 \end{pmatrix}$$

$$autovalores = \begin{pmatrix} 0.99546274 & 0.51906567 & 0.14012217 & 0.26293184 \end{pmatrix}$$

- **Etapa 5: Escolhendo as componentes e formando o vetor de características**

  Nesta etapa é onde ocorre a redução de dimensionalidade. Ao observar os autovetores e autovalores calculados anteriormente, é possível ver que os autovalores apresentam valores diferentes entre si. De fato, o autovetor com maior autovalor é a componente principal do *dataset*.

  De modo geral, uma vez obtida a matriz de covariância, o próximo passo é ordenar os autovalores por ordem de significância. Neste momento é possível ignorar as componentes de menor significância. Para o nosso exemplo, a Figura 14 apresenta a variância acumulativa para cada componente encontrada. É possível observar que as três primeiras componentes acumulam aproximadamente 92,69% da variância. A Tabela 7 apresenta os valores da variância acumulativa para cada componente. Como o objetivo é reduzir a dimensionalidade sem perder muita informação, a componente com menor variância é removida. Ao remover a componente 4, a matriz de autovetores fica da seguinte forma:

$$matriz_w = \begin{pmatrix} -0.72918013 & -0.67743377 & -0.07733456 \\ -0.62309911 & 0.6146496 & 0.4835888 \\ -0.27523283 & 0.35916489 & -0.81829251 \\ 0.06550358 & -0.18517545 & 0.30092958 \end{pmatrix}$$

  Uma vez escolhida as componentes (autovetores) que se deseja manter no problema, a matriz resultante será utilizada para obter o novo espaço de atributos. Para isso é necessário realizar a seguinte transformação :

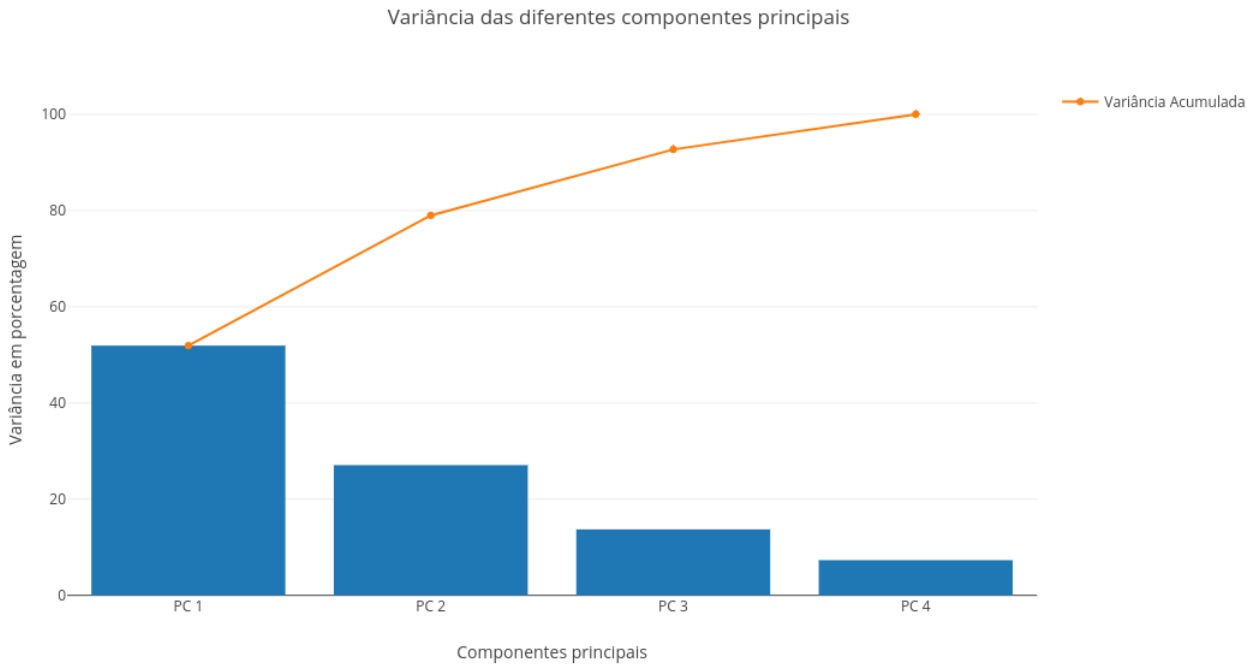

Figura 14 – Gráfico representando a variância acumulativa das componentes do problema.

Tabela 7 – Variância acumulada para cada componente do problema.

Eixo	Variância	Variância Acumulada
1	51,912383664561368%	51,912383664561368%
2	27,068754309873384%	78,981137974434752%
3	13,711631639030575%	92,692769613465327%
4	7,3072303865346813%	100%

Tabela 8 – Dados transformados pela aplicação do PCA com redução de uma dimensão.

Instâncias	Atributo 1	Atributo 2	Atributo 3
1	-1.51796862	0.19615918	-0.13186183
2	-1.45246505	0.01098373	0.16906775
3	-0.78878849	0.87359295	-0.05452727
4	0.56349075	0.93637712	-0.46078151
5	1.4618227	-0.03743737	-0.12607781
6	1.52732628	-0.22261282	0.17485177
7	0.79814614	-0.90004658	0.09751721
8	-0.89486951	-0.41849042	-0.61545063
9	0.00346244	-1.3923049	-0.28074693
10	0.83872359	0.57721223	0.357511
11	-0.5541331	-0.96283075	0.50377145
12	-0.1001858	0.0737679	-0.23718649
13	-0.51355566	0.51442806	0.76376524
14	0.62899433	0.75120167	-0.15985193

$$DadosFinal = dadosNormalizados * matriz_w$$

Onde *dadosNormalizados* refere-se ao conjunto de dados da Tabela 6 modelado em forma de matriz. Os dados gerados por esta transformação são apresentados na

Tabela 8.

3.3.2 Kernel Principal Component Analysis - KPCA

O PCA é uma técnica de projeção linear que trabalha bem com dados linearmente separáveis. Em caso de dados não-linearmente separáveis, uma técnica não-linear é preferível para reduzir a dimensionalidade.

A ideia básica para lidar com dados não-linearmente separáveis é projetá-los em um espaço dimensional maior para torná-lo linearmente separável. Para realizar essa tarefa é necessário uma função de mapeamento não-linear ϕ, de modo que o mapeamento de uma amostra x possa ser escrita como $x \rightarrow \phi(x)$, chamada de função kernel.

O termo "kernel"descreve uma função que calcula o produto escalar das imagens das amostras x sob ϕ, como demonstrado na Equação 3.12:

$$k(x_i x_j) = \phi(x_i)\phi(x_j)^T \tag{3.12}$$

Em outras palavras, a função ϕ mapeia um espaço d-dimensional de atributos em um espaço k-dimensional maior, criando combinações não-lineares dos atributos originais (WANG, 2012). Por exemplo, se x consiste em 2 atributos:

$$x = [x_1 \quad x_2]^T \qquad x \in \mathbb{R}^d$$

$$\Downarrow \phi$$

$$x\prime = [x_1 \quad x_2 \quad x_1x_2 \quad x_1^2x_1x2^3 \quad ...]^T \qquad x \in \mathbb{R}^k (k >> d)$$

Algumas das implementações de funções de kernel mais populares são a *linear kernel function*, *polynomial kernel function*, *Radial Basis Function* (RBF) kernel e a *Pearson VII universal kernel* , popularmente conhecida como PUK (ÜSTüN; MELSSEN; BUYDENS, 2006).

Na abordagem linear do PCA, as componentes que maximizam a variância são as mais interessantes para serem selecionadas. Isto é feito extraindo-se os autovetores(componentes principais) que correspondem aos maiores autovalores baseado na matriz de covariância:

$$Cov = \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T$$

Schölkopf, Smola e Müller (1999) generalizou essa abordagem para dados mapeados em espaços dimensionais maiores por meio do uso de função de kernel, como demonstrado na Equação 3.13:

$$Cov = \frac{1}{N}\sum_{i=1}^{N}\phi(x_i)\phi(x_i)^T \tag{3.13}$$

Na prática, a matriz de covariância em espaço dimensional maior não é calculada explicitamente, porque seria muito custoso, dessa forma são utilizadas as funções de kernel, mencionadas anteriormente, que fornecem uma projeção dos autovetores sobre as componentes principais. Na literatura, essa técnica é conhecida como *kernel trick*.

A implementação do kernel PCA pode ser sintetizada então em dois passos:

- **Passo 1: Computar a matriz kernel de similaridade**

  Para computar a matriz kernel é necessário calcular a similaridade (distância) entre os pares de pontos do *dataset*. A Tabela 9 apresenta os dados da Tabela 3 onde os atributos nominais foram transformados em binário (Expectativa (Exp), Temperatura (Tem), Humidade (Humidade), Vento (Ven)). Cada registro dessa tabela é presentado como um ponto do espaço N-dimensional, onde $N = 8$ (número de atributos sem contar o atributo Classe).

Tabela 9 – Dados da Tabela 3 com atributos binarizados.

Reg	Exp=Sol	Exp=Nublado	Exp=Chuva	Tem=Quente	Tem=Moderada	Tem=Fresco	Hum=Normal	Ven=Fraco	Classe
A	1	0	0	1	0	0	0	1	0
B	1	0	0	1	0	0	0	0	0
C	0	1	0	1	0	0	0	1	1
D	0	0	1	0	1	0	0	1	1
E	0	0	1	0	0	1	1	1	1
F	0	0	1	0	0	1	1	0	0
G	0	1	0	0	0	1	1	0	1
H	1	0	0	0	1	0	0	1	0
I	1	0	0	0	0	1	1	1	1
J	0	0	1	0	1	0	1	1	1
K	1	0	0	0	1	0	1	0	1
L	0	1	0	0	1	0	0	0	1
M	0	1	0	1	0	0	1	1	1
N	0	0	1	0	1	0	0	0	0

  A matriz de similaridade apresenta informações de quão similares ou distantes um do outro está cada par de pontos. Para realizar o cálculo são utilizadas funções que calculam a distância entre pontos, como a euclidiana. A distância entre os pontos $A = (a_1, a_2, a_3, ..., a_n)$ e $B = (b_1, b_2, b_3, ..., b_n)$ pode ser exemplificada pela Equação 3.14.

$$d_{AB}^2 = \sum_{i=1}^{n}(a_i - b_i)^2 \tag{3.14}$$

  Para calcular a matriz de similaridade é necessário calcular a distância euclidiana para cada par de pontos da matriz (*dataset*) (MxN). Para uma base de dados de m regis-

tros, há $((m-1) * m/2)$ pares, que são arranjados na ordem $(A, B), (A, C), ..., (A, m)$, $..., (B, m), ..., ..., (m-1, m)$. Os resultados são apresentados em uma matriz resultante MxM representada pela Tabela 10.

Tabela 10 – Tabela representando a matriz de similaridade dos dados da Tabela 9

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**
A	**0.**	1.	2.	4.	5.	6.	6.	2.	3.	5.	4.	5.	3.	5.
B	1.	**0.**	3.	5.	6.	5.	5.	3.	4.	6.	3.	4.	4.	4.
C	2.	3.	**0.**	4.	5.	6.	4.	4.	5.	5.	6.	3.	1.	5.
D	4.	5.	4.	**0.**	3.	4.	6.	2.	5.	1.	4.	3.	5.	1.
E	5.	6.	5.	3.	**0.**	1.	3.	5.	2.	2.	5.	6.	4.	4.
F	6.	5.	6.	4.	1.	**0.**	2.	6.	3.	3.	4.	5.	5.	3.
G	6.	5.	4.	6.	3.	2.	**0.**	6.	3.	5.	4.	3.	3.	5.
H	2.	3.	4.	2.	5.	6.	6.	**0.**	3.	3.	2.	3.	5.	3.
I	3.	4.	5.	5.	2.	3.	3.	3.	**0.**	4.	3.	6.	4.	6.
J	5.	6.	5.	1.	2.	3.	5.	3.	4.	**0.**	3.	4.	4.	2.
K	4.	3.	6.	4.	5.	4.	4.	2.	3.	3.	**0.**	3.	5.	3.
L	5.	4.	3.	3.	6.	5.	3.	3.	6.	4.	3.	**0.**	4.	2.
M	3.	4.	1.	5.	4.	5.	3.	5.	4.	4.	5.	4.	**0.**	6.
N	5.	4.	5.	1.	4.	3.	5.	3.	6.	2.	3.	2.	6.	**0.**

De posse da matriz de similaridade, já é possível aplicar a função kernel para obter a matriz kernel de similaridade. Para este exemplo foi utilizada a RBF kernel (Equação 3.15). Onde $||x_i - x_j||_2^2$ representa o cálculo de similaridade entre os pontos, e $\gamma = \frac{1}{2\sigma^2}$ é um parâmetro livre a ser otimizado.

$$\sum_{i=1}^{m}\sum_{j=1}^{m} k(x_i, x_j) = \sum_{i=1}^{m}\sum_{j=1}^{m} exp\left(-\gamma||x_i - x_j||_2^2\right) \tag{3.15}$$

Para o exemplo apresentado usado nesta demonstração, a função RBF kernel pode ser aplicada diretamente na matriz apresenta pela Tabela 10. O resultado desta operação com valor $\gamma = 2.0$ é apresentado na Tabela 11.

Tabela 11 – Matriz kernel de similaridade para a Tabela 9

	A	**B**	**C**	**...**	**L**	**M**	**N**
A	1.00000000e+00	1.35335283e-01	1.83156389e-02	...	4.53999298e-05	2.47875218e-03	4.53999298e-05
B	1.35335283e-01	1.00000000e+00	2.47875218e-03	...	3.35462628e-04	3.35462628e-04	3.35462628e-04
C	1.83156389e-02	2.47875218e-03	1.00000000e+00	...	2.47875218e-03	1.35335283e-01	4.53999298e-05
...	...	...	...	...	...	...	...
L	4.53999298e-05	3.35462628e-04	2.47875218e-03	...	1.00000000e+00	3.35462628e-04	1.83156389e-02
M	2.47875218e-03	3.35462628e-04	1.35335283e-01	...	3.35462628e-04	1.00000000e+00	6.14421235e-06
N	4.53999298e-05	3.35462628e-04	4.53999298e-05	...	1.83156389e-02	6.14421235e-06	1.00000000e+00

- **Passo 2: Autodecomposição da matriz de kernel:**

  Como não é garantido que a matriz kernel está centralizada, é aplicada a Equação 3.16, para torná-la:

$$K^{'} = K - 1_N K - K 1_N + 1_N K 1_N \tag{3.16}$$

Onde 1_N é uma matriz NxN com todos os valores iguais a $\frac{1}{N}$. O resultado da centralização é apresentada na Tabela 12.

Tabela 12 – Matriz kenel centralizada

	A	**B**	**C**	**...**	**L**	**M**	**N**
A	0.91242068	0.05015075	-0.06797938	...	-0.07710749	-0.08255549	-0.08770773
B	0.05015075	0.91721025	-0.08142148	...	-0.07442264	-0.082304	-0.08502288
C	-0.06797938	-0.08142148	0.91498928	...	-0.07338984	0.05158534	-0.08642343
...	...	...	...	...	...	...	...
L	-0.07710749	-0.07442264	-0.07338984	...	0.93327354	-0.07427235	-0.05901106
M	-0.08255549	-0.082304	0.05158534	...	-0.07427235	0.91751083	-0.08520191
N	-0.08770773	-0.08502288	-0.08642343	...	-0.05901106	-0.08520191	0.91207306

Depois dessa etapa são calculados os autovalores e autovetores sobre a matriz de kernel centralizada, a partir daqui o algoritmo do KPCA funciona como o PCA. Os autovetores que apresentam os maiores autovalores são selecionadas como componentes principais. Abaixo são apresentados os autovalores calculados a partir da Tabela 12 em ordem decrescente de magnitude:

$$autovalores = \begin{pmatrix} 1.18064493e+00 & 1.14043249e+00 & 1.12468024e+00 \\ 1.05088323e+00 & 1.00926291e+00 & 1.00092583e+00 \\ 9.98339041e-01 & 9.82113290e-01 & 9.75872071e-01 \\ 8.70889930e-01 & 8.62296155e-01 & 8.57013341e-01 \\ 8.17179259e-01 & 1.11022302e-15 & \end{pmatrix}$$

Como pode ser observado, as primeiras 8 componentes acumulam cerca de 65% da variância. Assim sendo, são selecionados os 8 primeiros autovetores para compor a base final representada pela Tabela 13. Esses autovetores são pontos já projetados nas respectivas componentes principais.

Tabela 13 – Resultado da aplicação do RBF Kernel PCA na Tabela 9

	PC1	**PC2**	**PC3**	**PC4**	**PC5**	**PC6**	**PC7**	**PC8**
1	-2.08023856e-01	-2.89075725e-01	4.98305445e-01	-2.29325089e-01	-1.15202831e-03	-7.30282646e-04	-4.32291034e-07	5.29954535e-05
2	-2.07284266e-01	-2.88101478e-01	4.99981415e-01	-2.29924108e-01	2.31079533e-03	1.94243011e-04	-2.18237739e-07	-4.75400267e-03
3	-2.07451341e-01	-2.88061567e-01	-5.00007154e-01	-2.28196328e-01	-1.22764238e-03	-7.65344393e-04	-3.22125879e-05	2.08315645e-05
4	6.10850117e-01	-5.88869890e-03	-2.28290468e-05	-1.80518578e-01	-2.86495967e-03	1.41169403e-03	6.41876264e-05	5.29900331e-05
5	-2.00250251e-01	5.77893043e-01	3.10784267e-05	-2.32458764e-01	-3.12243607e-03	-3.83372450e-03	-4.74881346e-03	2.15275316e-07
6	-2.01495160e-01	5.77897625e-01	3.10779874e-05	-2.31921929e-01	-1.97413964e-03	-5.67502060e-03	4.68155399e-03	5.29874229e-05
7	-4.78066338e-02	4.55215591e-03	-2.27467212e-05	3.51240331e-01	2.80875421e-01	4.54139742e-01	7.09499113e-01	3.09134078e-03
8	-4.48716285e-02	-1.34459318e-03	3.39182578e-03	3.54395369e-01	-5.23701675e-01	-1.41764138e-01	4.77816088e-03	7.08626838e-01
9	-4.78018410e-02	4.55188705e-03	2.31141187e-05	3.51272312e-01	2.63242114e-01	4.71995285e-01	-7.04459930e-01	3.09104238e-03
10	4.25819680e-01	-1.35200785e-03	-2.27988092e-05	-1.11119497e-01	-1.53650199e-01	2.55954539e-01	4.84317328e-03	-7.84357349e-03
11	-4.69552798e-02	6.27442264e-04	4.59066207e-05	3.61322882e-01	-5.21259988e-01	-1.42302371e-01	4.74682504e-03	-7.05503519e-01
12	-4.45138047e-02	6.01129186e-04	-2.27287367e-05	3.53534941e-01	5.07820606e-01	-6.32201435e-01	-1.44969130e-02	1.39070798e-06
13	-2.06840981e-01	-2.87088135e-01	-5.01688751e-01	-2.27086336e-01	-1.23006886e-03	-7.11298373e-04	2.22804304e-07	2.06108981e-05
14	4.26625246e-01	-5.21107698e-03	-2.28547633e-05	-1.01215206e-01	1.55934200e-01	-2.55711889e-01	-4.87471732e-03	3.08985239e-03

3.3.3 Independent Component Analysis - ICA

Medições em que a fonte de sinal não pode ser distinguida do ruído não são interessantes para classificação de sinais. Por exemplo, o som de uma pessoa gravada em uma rua, sempre será corrompida por outros sons do ambiente. Isso acontece porque em um

ambiente há várias fontes de sinais diferentes que estão sendo gravadas junto com os sons produzidos pelas pessoas. Dessa forma, o som obtido pela gravação será uma combinação de várias fontes independentes de sons. Problemas que buscam separar as fontes de sinais de um conjunto de sinais combinados é conhecido como *blid source separation* (BSS). O termo *blind* (cego) indica que os sinais podem ser separados mesmo que somente uma pequena informação das fontes de sinais seja conhecida (STONE, 2002).

Uma das mais amplamente conhecidas técnicas que resolvem esse tipo de problema é o *Independent Component Analysis* (ICA). O objetivo do ICA é extrair informações úteis dos dados ou fontes de sinais dos dados (conjunto de combinações de sinais). Esses dados podem ser na forma de imagens, sons, informações sobre mercado de ações e etc. ICA também é considerado um algoritmo de redução de dimensionalidade porque pode separar as fontes de sinais em componentes independentes que podem ser removidas dependendo da necessidade (COMON, 1994).

O ICA é baseado em um modelo gerador que assume algumas fontes de sinais independentes (s) são combinados linearmente com uma matriz de combinação A:

$$x = As \tag{3.17}$$

A Equação 3.17 implica que se a inversa de A (Equação 3.18) multiplicada pelos sinais x é possível obter as fontes (Equação 3.19):

$$W = A^{-1} \tag{3.18}$$

$$s = xW \tag{3.19}$$

Isto significa que a principal tarefa do ICA é estimar o valor de W.

Para que o ICA possa funcionar corretamente é necessário que três condições sejam obedecidas: A primeira é que os sinais x observados sejam uma combinação linear de componentes independentes. Independência significa que o sinal x_1 não contém nenhuma informação do sinal x_2. Segundo, os sinais não podem ser correlacionados e precisam ter covariância igual a zero. A terceira condição que precisa ser cumprida é que as fontes de sinais independentes tenham distribuição não-gaussiana. Isso porque a densidade de distribuição gerada pela combinação dos sinais independentes não-gaussianos será uniforme.

O cálculo do ICA pode ser sintetizado em duas etapas: Etapa de pré-processamento e a etapa de extração das componentes independentes. Abaixo as etapas são detalhadas por meio de um exemplo.

- **Passo 1: Pré-processamento**

Inicialmente é realizada uma centralização dos dados, é a mesma normalização realizada no PCA, onde é feita uma subtração simples da média do valor de entrada X $(x - \overline{x})$. Depois é realizado outro pré-processamento chamado *whitening*. O objetivo é transformar linearmente os sinais observados X de maneira remover o máximo possível a correlação entre eles e igualar suas variâncias a zero. Como resultado a matriz de covariância resultante será igual a matriz identidade. Esse processo é descrito abaixo usando a Tabela 6, cujo os dados já se encontram normalizados.

Com os dados normalizados é dado início ao processo de *whitening*, que apresenta as seguintes subetapas: calculo da matriz de covariância(Equação 3.11), decomposição em valores singulares (Single Value Decomposition - SVD) (Equação 3.20), calculo da matriz diagonal de autovalores (Equação 3.22), cálculo da matriz *whitening* (Equação 3.23) e projeção da matriz *whitening sobre os dados* (Tabela 14)

A matriz de covariância (Cov) é a mesma calculada no exemplo do PCA. Na Equação 3.20, tem-se que uma matriz Cov (mxn) pode é decomposta em três matrizes ($U(m$x$n)$, Σ (mxn), V^T (nxm)). O resultado da decomposição pode ser visto logo abaixo:

$$Cov = U.\Sigma.V^T \tag{3.20}$$

$$\Downarrow$$

$$Cov = \begin{pmatrix} -0.7291 & 0.6774 & -0.0583 & 0.0773 \\ -0.6230 & -0.6146 & 0.0097 & -0.4835 \\ -0.2752 & -0.3591 & 0.3544 & 0.8182 \\ 0.0655 & 0.1851 & 0.9331 & -0.3009 \end{pmatrix} * \begin{pmatrix} 0.9954 & 0. & 0. & 0. \\ 0. & 0.5190 & 0. & 0. \\ 0. & 0. & 0.2629 & 0. \\ 0. & 0. & 0. & 0.1401 \end{pmatrix} * \begin{pmatrix} -0.7291 & -0.6230 & -0.2752 & 0.0655 \\ 0.6774 & -0.6146 & -0.3591 & 0.1851 \\ -0.0583 & 0.0097 & 0.3544 & 0.9331 \\ 0.0773 & -0.4835 & 0.8182 & -0.3009 \end{pmatrix} \tag{3.21}$$

Com a matriz decomposta, é calculada a matriz diagonal de autovalores conforme demonstrada na Equação 3.22. Onde σ_i são os valores da diagonal da matriz Σ retornada pela decomposição.

$$d = \begin{pmatrix} \frac{1}{\sqrt{\sigma_1}} & 0. & 0. & 0. \\ 0. & \frac{1}{\sqrt{\sigma_2}} & 0. & 0. \\ 0. & 0. & \frac{1}{\sqrt{\sigma_3}} & 0. \\ 0. & 0. & 0. & \frac{1}{\sqrt{\sigma_4}} \end{pmatrix} = \begin{pmatrix} 1.0022 & 0. & 0. & 0. \\ 0. & 1.3879 & 0. & 0. \\ 0. & 0. & 1.9501 & 0. \\ 0. & 0. & 0. & 2.6714 \end{pmatrix} \tag{3.22}$$

Obtida a matriz diagonal de autovalores já é possível calcular a matriz *whitening* (Mw). A Equação 3.23 apresenta o cálculo da matriz Mw a partir das matrizes U e d.

$$Mw = U.d.U^T = \begin{pmatrix} 1.19249527 & -0.22357167 & -0.00781104 & -0.04203326 \\ -0.22357167 & 1.53843917 & -0.57208822 & 0.20763873 \\ -0.00781104 & -0.57208822 & 2.28881665 & -0.12312597 \\ -0.04203326 & 0.20763873 & -0.12312597 & 1.99216709 \end{pmatrix} \tag{3.23}$$

Calculada a matriz de transformação Mw, é realizada a sua projeção sobre os dados da Tabela 6. O resulta dessa projeção pode ser observado na Tabela 14

Tabela 14 – Resultado da projeção da matriz whitening sobre os dados da Tabela 6

Expectativa	**Temperatura**	**Humidade**	**Vento**
0.98267205	0.94161512	0.61622198	-0.73266769
0.94063879	1.14925385	0.49309601	1.2594994
-0.20982322	1.1651868	0.62403302	-0.69063444
-1.17874682	-0.1496807	1.20393227	-0.85623991
-0.94736411	-1.11603166	-0.51279616	-0.94075266
-0.98939737	-0.90839293	-0.63592214	1.05141443
0.2030979	-1.1319646	-0.64373317	1.00938117
1.20624372	-0.59682405	1.1883102	-0.94030642
1.43762643	-1.563175	-0.52841824	-1.02481917
-1.17093578	0.42240751	-1.08488438	-0.73311394
1.1720215	0.1829029	-1.22363243	1.17498664
-0.02828481	-0.16561365	1.07299526	1.09389393
-0.20201219	1.73727501	-1.66478363	-0.56750846
-1.22078008	0.05795803	1.0808063	1.13592718

- **Passo 2: Extraindo as componentes independentes**

  Uma das condições para o algoritmo ICA funcionar é que os sinais fontes sejam não-gaussianos. Assim sendo, a busca pelas componentes independentes pode ser alcançada pela maximização da não-gaussianidade dos sinais extraídos. Duas medidas utilizadas para calcular a não-gaussianidade são, *kurtosis* e a entropia negativa (STONE, 2002). Neste exemplo, será utilizada a abordagem fastICA (Hyvarinen, 1999) que utiliza entropia negativa para maximizar a não-gaussianidade utilizando um esquema de iteração de ponto-fixo.

$$g(u) = tanh(u) \tag{3.24}$$

$$g'(u) = 1 - tanh^2(u) \tag{3.25}$$

  No algoritmo de iteração de ponto fixo (FastICA), são utilizadas as Equações 3.24 e 3.25 para fazer uma aproximação da entropia negativa. O pseudocódigo que representa esse algoritmo pode ser observado logo abaixo.

  De acordo com o algoritmo, os pesos são atualizados até que convergirem a um limite (*threshold*). Inicialmente os pesos(w_p) para cada componente são inicializados randomicamente. O produto dos pesos randômicos(W^T) com os sinais combinados (X que é o mesmo Mw retornado pela etapa anterior) passam pelas funções $g(W^T X)$ e $g'(W^T X)$. Então é calculada a média da subtração de g' e g. Para impedir que na iteração da segunda componente seja identificado a mesma componente da primeira iteração, é realizado um calculo de decorrelação dos novos pesos em relação aos pesos anteriores. Em seguida, os novos pesos são divididos pela sua norma.

Algorithm 1 fastICA

for 1 *até o número de componentes* **c do**
  $w_p \leftarrow$ *inicialização randômica*
  while w_p $>$ *threshold* **do**
    $w_p \leftarrow \frac{1}{n}(Xg(W^TX) - g'(W^TX)W)$
    $w_p \leftarrow w_p - \sum_{j=1}^{p-1}(w_p^T w_j)w_j$
    $w_p \leftarrow \frac{w_p}{||w_p||}$
  end while
  $W \leftarrow [w_1, ..., w_c]$
end for

Depois que é calculado os pesos para cada componente, é retornada a matriz W, que é a matriz inversa aproximada utilizada para obter os sinais independentes da matriz original (Equação 3.18). A matriz W retornada pelo FastICA com threshold igual a 1e-8 e 4 componentes independentes é apresentado abaixo:

$$W = \begin{pmatrix} -0.01206244 & 0.12442531 & -0.02273088 & -0.99189523 \\ 0.47611612 & -0.65986059 & 0.57232736 & -0.10168008 \\ -0.05628268 & 0.62393465 & 0.77704513 & 0.06114481 \\ -0.87749655 & -0.3997598 & 0.26100876 & -0.04545687 \end{pmatrix}$$

O resultado da Tabela 14 retornada pela etapa de pré-processamento (matriz Mw) é multiplicado pela matriz W para obter as correspondentes componentes independentes dos dados originais. O resultado é apresentado pela Tabela 15.

Tabela 15 – Resultado do FastICA sobre os dados da Tabela 14

CI_1	**CI_2**	**CI_3**	**CI_4**
0.81802966	0.2737097	0.96623234	-1.0445671
-1.12885008	-0.1563477	1.12428733	-1.21338331
0.81836189	-0.44138613	1.18148293	-0.08740353
0.81752833	0.31365395	0.85610713	1.44734145
0.81734932	0.08753852	-1.09899853	1.18637271
-1.12953041	-0.34251888	-0.94094354	1.0175565
-1.12986265	0.37257695	-1.15619413	0.06039293
0.81686386	1.74384561	0.42560596	-0.46698568
0.81668484	1.51773017	-1.5294997	-0.72795442
0.81851512	-1.3825974	-0.55837215	0.6087913
-1.12902909	-0.38246314	-0.83081833	-1.47435205
-1.12968364	0.59869238	0.79891153	0.32136167
0.81934868	-2.13763748	-0.23299635	-0.92595368
-1.1293514	-0.11640344	1.01416212	1.27852524

3.3.4 Latent Semantic Analysis - LSA

O LSA, também conhecido por LSI (Latent Semantic Indexing), é uma extensão do método de recuperação de vetores (SALTON; MCGILL, 1983) em que as dependências

entre os termos e documentos, além das associações entre termos e documentos, são explicitamente levadas em conta. Em outras palavras, o LSA analisa os documentos em busca de significados ou conceitos escondidos. Surgiu a partir do problema de encontrar documentos relevantes a partir de palavras de busca. A dificuldade fundamental aparece quando é necessário comparar palavras para encontrar documentos relevantes, porque o que realmente se quer comparar são significados ou conceitos por traz das palavras. LSA tenta resolver esse problema mapeando palavras e documentos dentro de um espaço de conceito e fazendo comparações neste espaço.

O primeiro passo para se aplicar o LSA é transformar os dados em uma matriz de frequência, onde as palavras são os atributos e os documentos são cada um dos registros da base de dados. Cada célula da matriz conterá o número de vezes que aquela palavra apareceu naquele registro. Em geral, é comum que as matrizes resultantes dessa transformação sejam largas(com muitos atributos) e muito esparsa (com muitas células contendo 0). É comum também que após a transformação dos dados em uma matriz de frequência, seja aplicado um algoritmo de ponderação sobre os dados. A técnica de ponderação mais comumente utilizada para esse fim é o TF-IDF.

Depois de ponderada, a matriz resultante é decomposta utilizando SVD. A decomposição SVD permite reduzir a dimensionalidade da matriz, enfatiza os relacionamentos mais importantes e remove os ruídos. Em outras palavras, faz a melhor reconstrução possível da matriz com o mínimo de informação. Abaixo é apresentado um exemplo de aplicação do LSA sobre os dados da Tabela 9.

- **Passo 1: Transformação da base**

  Inicialmente, é necessário realizar uma transformação da base para que ela obedeça a estrutura (documento, termo) $\rightarrow$ (registro, atributo), onde cada elemento da estrutura represente a quantidade de vezes que o termo aparece em um documento. Para o exemplo da Tabela 9, tem-se que os atributos (Sol,Chuva,Nublado) antes categorias do atributo Expectativa, passam a ser termos na nova representação da base. O mesmo acontece com os atributos Temperatura(Quente, Moderada, Fresco), Humidade(Normal) e Vento(Fraco).

  Após esta transformação inicial é aplicada a Equação 3.3 sobre os dados da tabela. O resultado pode ser observado na Tabela 16

- **Passo 2: Decomposição em valores singulares**

  Nesta etapa é aplicada a decomposição SVD(Equação 3.20) sobre a matriz resultante da etapa anterior. Como o objetivo é reduzir a dimensionalidade, a decomposição selecionará os k maiores valores singulares de Σ. Dessa forma uma matriz B aproximada

Tabela 16 – Resultado da aplicação da representação TF-IDF sobre os dados da Tabela 9

Expectativa=Sol	Expectativa=Nublado	Expectativa=Chuva	Temperatura=Quente	Temperatura=Moderada	Temperatura=Fresco	Humidade=Normal	Vento=Fraco
0.15	0.	0.	0.26	0.	0.	0.	1.
0.26	0.	0.	0.5	0.	0.	0.	0.
0.	0.18	0.	0.32	0.	0.	0.	1.
0.	0.	0.15	0.	0.17	0.	0.	1.
0.	0.	0.13	0.	0.	0.17	0.13	1.
0.	0.	0.21	0.	0.	0.29	0.24	0.
0.	0.21	0.	0.	0.	0.34	0.27	0.
0.2	0.	0.	0.	0.19	0.	0.	1.
0.18	0.	0.	0.	0.	0.28	0.2	1.
0.	0.	0.19	0.	0.16	0.	0.24	1.
0.29	0.	0.	0.	0.26	0.	0.43	0.
0.	0.38	0.	0.	0.51	0.	0.	0.
0.	0.22	0.	0.26	0.	0.	0.3	1.
0.	0.	0.46	0.	0.53	0.	0.	0.

da matriz original A pode ser reconstruída como a equação abaixo:

$$B = U.\Sigma_k.V_k^T \tag{3.26}$$

Na prática, pode-se reter e tratar com um subconjunto descritivo de dados chamado de M. Que é um resumo denso da matriz ou projeção:

$$M = U.\Sigma_k \tag{3.27}$$

Essa matriz de Transformação M pode ser calculada e aplicada a matriz original A assim como em outras matrizes similares:

$$M = V_k^T.A \tag{3.28}$$

Para um valor $k = 2$, tem-se a matriz resultante $M(m\text{x}2)$ representada pela Tabela 17.

Tabela 17 – Resultado final da projeção da matriz M para 2 componentes

Component_1	Component_2
-1.01538563	0.1525291
-0.07203987	0.04025835
-1.02108516	0.13349273
-0.99871993	-0.0644679
-1.01020131	0.02259618
-0.0582054	-0.18877082
-0.06292173	-0.1644854
-1.00481819	-0.0370527
-1.02935888	0.02648603
-1.02831176	-0.14412088
-0.08922965	-0.36216392
-0.05796942	-0.49342656
-1.05157222	0.02782685
-0.0675772	-0.61206097

3.4 Classificadores

Depois de definido os subconjuntos de atributos, é necessário avaliar o quanto a escolha de um subconjunto em específico pode impactar no processo de aprendizagem.

Para se inferir o quanto um conjunto de atributos se saiu em comparação a outro, quando o atributo classe é conhecido, é comum o uso de algoritmos conhecidos na literatura como Classificadores. Esta técnica, classificação, faz parte de uma subcategoria da Aprendizagem de Máquina Indutiva, que é a Aprendizagem de Máquina Supervisionada. A tarefa de classificação consiste em dividir os dados em grupos de classes. Os registros são agrupados de acordo com um atributo alvo (classe), que determina em qual região de decisão está o registro. Segundo (ARAUJO, 2014) o objetivo é descobrir as relações existentes entre o atributos de predição e o atributo alvo, utilizando registros cuja classificação é conhecida. Existem diversos paradigmas para aprendizado supervisionado (MITCHELL, 1997), neste trabalho serão abordados quatro algoritmos para cada um dos paradigmais: simbólico (*Random Forest*), estatístico (*Naive Bayes*), baseado em exemplos (*K-nearest neighbors*) e conexionista (Support Vector Machine).

3.4.1 RF - *Random Forest*

O algoritmo *Random Forest* pode ser entendido como uma combinação de diversas árvores de decisão, onde cada árvore é gerada a partir de amostras selecionadas aleatoriamente do conjunto de dados da base. Após a geração de um grande número de árvores, as classes com maior número de votos são eleitas.

Para melhor exemplificar, supondo um conjunto de dados S cujos valores podem ser descritos numa matriz como a apresentada abaixo:

$$S = \begin{pmatrix} a_{X1} & a_{Y1} & a_{Z1} & C_1 \\ \vdots & \vdots & \vdots & \vdots \\ a_{XN} & a_{YN} & a_{ZN} & C_N \end{pmatrix}$$

Onde a_{X1} significa o valor do atributo X na amostra 1, a_{Y1} o valor do atributo Y na amostra 1 e assim sucessivamente. O valor C diz respeito ao atributo classe daquela amostra específica. O algoritmo RF gera M subconjuntos aleatórios com base S, como os apresentados a seguir:

$$S_1 = \begin{pmatrix} a_{X12} & a_{Y12} & a_{Z12} & C_{12} \\ a_{X15} & a_{Y15} & a_{Z15} & C_{15} \\ \vdots & \vdots & \vdots & \vdots \\ a_{X35} & a_{Y35} & a_{Z35} & C_{35} \end{pmatrix}$$

$$S_2 = \begin{pmatrix} a_{X2} & a_{Y2} & a_{Z2} & C_2 \\ a_{X6} & a_{Y6} & a_{Z6} & C_6 \\ \vdots & \vdots & \vdots & \vdots \\ a_{X20} & a_{Y20} & a_{Z20} & C_{20} \end{pmatrix}$$

$$S_M = \begin{pmatrix} a_{X4} & a_{Y4} & a_{Z4} & C_4 \\ a_{X9} & a_{Y9} & a_{Z9} & C_9 \\ \vdots & \vdots & \vdots & \vdots \\ a_{X12} & a_{Y12} & a_{Z12} & C_{12} \end{pmatrix}$$

Como o RF é um combinador de árvores de decisão, cada subconjunto gerado (S_1, S_2, S_M) é utilizado para criar uma árvore de decisão diferente, independente das outras. A classificação ocorre por votação, onde uma instância de entrada é classificada por cada árvore do RF e o resultado final é dado para a classe com o maior número de votos. A Figura 15 apresenta uma síntese desse algoritmo.

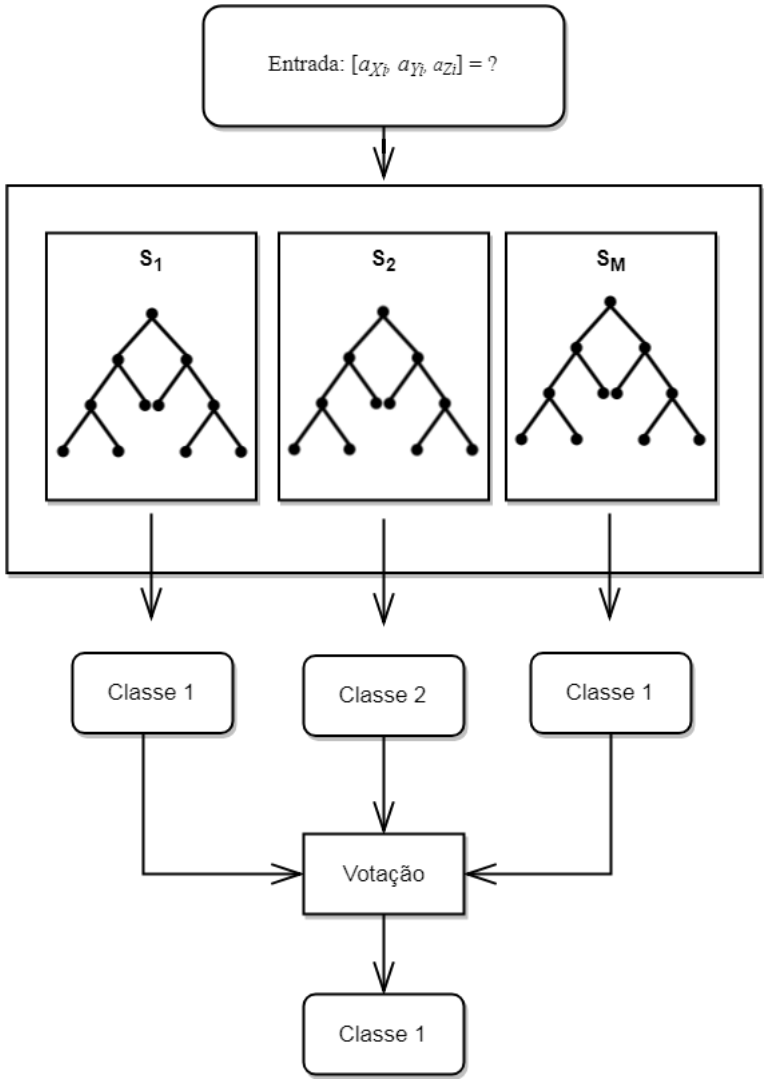

Figura 15 – Representação gráfica do algoritmo *Random Forest*

3.4.2 NB - *Naïve Bayes*

Este método tem esse nome devido ao fato de ser baseado no teorema de *Bayes* e "inocentemente"assume a independência entre os atributos (JOHN; LANGLEY, 1995). É

um algoritmo simplista mas que funciona bem com vários tipos de bases, particularmente quando combinado com algoritmos de seleção de atributos que eliminam redundância e atributos interdependentes (WITTEN et al., 2016).

O teorema de *Bayes*, a qual este algoritmo tem inspiração, funciona da seguinte forma: dada uma nova instância $A = a_1, a_2, ..., a_n$, deseja-se predizer sua classe assim como na Equação 3.29:

$$P(classe|a_1, a_2, ..., a_n) = \frac{P(a_1, a_2, ..., a_n|classe) * P(classe)}{P(a_1, a_2, ..., a_n)} \tag{3.29}$$

Segundo Araujo (2014), para se calcular a classe mais provável para uma determinada instância, deve-se calcular a probabilidade de todas as classes e, por fim escolhe-se a classe de maior probabilidade como rótulo da nova instância. Estatisticamente falando este cálculo equivale a maximizar a probabilidade $P(classe|a_1, a_2, ..., a_n)$. Assim sendo, deve-se maximizar o valor de $P(a_1, a_2, ..., a_n|classe) * P(classe)$ e minimizar o valor de $P(a_1, a_2, ..., a_n)$. Como o valor de $P(a_1, a_2, ..., a_n)$ é uma constante, que não depende do valor da classe que se está sob análise, pode-se anulá-lo, assim a nova equação fica da seguinte forma:

$$argmax P(classe|a_1, a_2, ..., a_n) = P(a_1, a_2, ..., a_n|classe) * P(classe) \tag{3.30}$$

Como o algoritmo NB supõe que os atributos sejam independentes, o cálculo de $P(a_1, a_2, ..., a_n|classe)$ é reduzido para $P(a_1|classe)*, ..., P(a_n|classe)$. Assim sendo a equação final fica da seguinte forma:

$$argmax P(classe|a_1, a_2, ..., a_n) = argmax \prod_i P(a_i|classe) * P(classe) \tag{3.31}$$

3.4.3 SVM - *Support Vector Machine*

O SVM é uma técnica proposta por (CORTES; VAPNIK, 1995) que realiza um mapeamento do espaço de entrada para um de dimensionalidade maior. Diz-se que duas classes são linearmente separáveis se existe um hiperplano que as separe. Sendo assim, esta técnica busca calcular um hiperplano de separação ótimo que maximize a distância de separação entre as classes (HAYKIN, 1994).

Ao utilizar SVM para reconhecimento de padrões se faz necessário tornar uma função não-linearmente separável em uma função linearmente separável. Para isso é necessário que seja aumentada a dimensionalidade do problema (COVER, 1965). As funções que aumentam a dimensionalidade do espaço de entrada são chamadas funções de Kernel.

Ao aplicar uma função Kernel em um vetor de entrada de dimensão N, é obtido um novo vetor de dimensão X, onde $X > N$. Depois disso são calculados os vetores de suporte. Com os vetores calculados é possível definir o hiperplano de separação ótimo. O hiperplano estará a uma igual distância (d_0) dos vetores de suporte da classe. Um novo objeto é classificado utilizando o hiperplano ótimo, onde cada lado do hiperplano representa uma classe diferente.

3.4.4 KNN - *K-nearest neighbors*

O KNN é um dos algoritmos de aprendizagem mais simples. É baseado na ideia de que objetos que estão próximos um dos outros possuem características similares. Então se é conhecido as características de um dos objetos, é possível predizer o mesmo para sua vizinhança.KNN é uma versão estendida da técnica NN (*Nearest Neighbors*) onde qualquer nova instância pode ser classificada pelo voto da maioria de seus k vizinhos - onde k é um inteiro positivo, frequentemente um número pequeno (FIX; JR, 1951).

Mesmo sendo uma técnica simples, o custo computacional é elevado. Do ponto de vista em consumo de memória, o KNN armazena todos os padrões de treinamento, e quanto ao desempenho, para cada padrão a ser classificado o KKN deve calcular sua distância para todos os padrões de treinamento (ARAUJO, 2014).

3.5 Métodos de Avaliação

A fim de avaliar os resultados obtidos por um classificador, é comum utilizar métodos de avaliação. Na literatura, a maioria das métricas de avaliação partem da Matriz de Confusão.

A matriz de confusão é uma tabela que mostra o resultado da classificação, indicando a quantidade classificações corretas e incorretas. Para um problema de classificação binário, a matriz é composta por 4 valores: Verdadeiro Positivo (VP), Verdadeiro Negativo (VN), Falso Positivo (FP) e Falso Negativo (FN) (CHIMIESKI; FAGUNDES, 2013). Para o trabalho de classificação de requisições Autorizadas e Não-autorizadas no processo de regulação, essas quatro variáveis podem ser definidas como:

- VP: número de procedimentos/exames/tratamentos que são corretamente classificados como Não-autorizados;
- FP: número de procedimentos/exames/tratamentos classificados como Não-autorizados, quando na verdade eram autorizados;
- FN: número de procedimentos/exames/tratamentos classificados como Autorizado, mas que na verdade eram Não-autorizados;

- VN: número de procedimentos/exames/tratamentos classificados corretamente como Autorizados.

A partir desses valores é possível calcular outras métricas, tais como: Precisão (P), *Recall*(R), Acurácia(A), Precision-Recall Curve (PRC) e índice Kappa (K) (GOMES, 2002).

- **Precisão (P):** reflete a proporção de verdadeiros positivos em relação a todas as predições positivas. A precisão de uma classificação mostra a quantidade de objetos da classe X classificados corretamente em relação a todos os objetos classificados como sendo da classe X. Ela pode ser calculado com a Equação 3.32.

$$P = \frac{VP}{VP + FP} \tag{3.32}$$

- **Recall (R):** reflete a proporção de verdadeiros positivos em relação a suas predições positivas e as suas incorretas predições negativas. Essa medida mostra o comportamento dos objetos classificados como X, ou seja, de todos os objetos da classe X, quantos foram classificados como X. A Equação 3.33 demonstra como é realizado este cálculo.

$$R = \frac{VP}{VP + FN} \tag{3.33}$$

- **Acurácia (A):** é a porcentagem de casos corretamente classificados em um conjunto de teste. Essa medida mede o quão bem um classificador reconhece instâncias de diversas classes, sendo calculada conforme a Equação 3.34.

$$A = \frac{VP + VN}{VP + FP + FN + VN} \tag{3.34}$$

- **Precision-Recall Curve (PRC):** é uma representação cartesiana dos valores das predições positivas (Precisão - eixo Y) contra a taxa de verdadeiros positivos (Recall - eixo x). Essa métrica de avaliação é comumente utilizada quando se deseja avaliar os resultados do classificador em base de dados desbalanceadas. Uma boa avaliação de um classificador, segundo a PRC, seria aquela em que a curva gerada mais se aproxime do ponto (1, 1) do plano cartesiano. Uma outra análise que pode ser realizado ao observar o gráfico é a área sob a curva PRC. Quando mais próximo de 1, melhor o resultado da classificação.

- **Índice Kappa (K):** é utilizado como medida de exatidão por representar inteiramente a matriz de confusão. Este índice toma todos os elementos da matriz em

consideração, em vez de apenas aqueles que retratam a quantidade de classificações verdadeiras, o que ocorre quando de calcula a exatidão global da classificação (ROSENFIELD; FITZPATRICK-LINS, 1986).

O índice Kappa é um coeficiente de concordância para escalas nominais que mede o relacionamento entre a concordância, além da casualidade, e a discordância esperada(ROSENFIELD; FITZPATRICK-LINS, 1986). Pode ser encontrado com base na Equação:

$$K = \frac{\theta_1 - \theta_2}{1 + \theta_2} \tag{3.35}$$

onde,

$$\theta_1 = \frac{VP + VN}{VP + VN + FP + VN} \tag{3.36}$$

e

$$\theta_2 = \frac{[(VP + FN)(VP + FP)] + [(VN + FN)(VN + FP)]}{(VP + VN + FP + FN)^2} \tag{3.37}$$

Neste caso, θ_1 é o valor global para a porcentagem correta, ou seja, o somatório da diagonal principal da matriz dividido pela quantidade de elementos e θ_2 são os valores calculados usando-se os totais de cada linha e cada coluna da matriz de confusão. A qualidade dos resultados podem ser avaliados pelo valor do índice Kappa observado na Tabela 18(LANDIS; KOCH, 1977).

Tabela 18 – nível de exatidão de uma classificação, conforme valor de índice Kappa

Kappa Index(K)	**Quality**
K $\leq$ 0.2	Ruim
0.2 $\leq$ K $\leq$ 0.4	Razoável
0.4 $\leq$ K $\leq$ 0.6	Bom
0.6 $\leq$ K $\leq$ 0.8	Muito Bom
K $\geq$ 0.8	Excelente

3.5.1 Teste de Hipótese Z

O teste de hipótese Z (CONGALTON; GREEN, 2008) é feito utilizando o índice *Kappa* como métrica, pois é o mais comumente utilizada para esse fim. O objetivo é testar a significância estatística da diferença entre dois índices *Kappa*, como descrito na Equação 3.38:

$$Z = \frac{K_1 - K_2}{\sqrt{\sigma_{k2}^2 + \sigma_{k1}^2}} \tag{3.38}$$

Onde K_1 = índice *Kappa* da classificação 1; K_2 = índice *Kappa* da classificação 2 e σ^2 =variância do índice *Kappa*, que pode ser calculado por:

$$\sigma^2 = \frac{\theta_1(1-\theta_1)}{N(1-\theta_2)^2}, \tag{3.39}$$

onde N é o número total de elementos da matriz de confusão e θ_1 e θ_2 são calculados conforme Equação 3.36 e Equação 3.37, respectivamente. O objetivo do teste Z é avaliar uma determinada hipótese dentro de uma zona de aceitação ou rejeição. Dessa forma foram definidas as Hipóteses nula(H_0) e alternativa (H_1) para o teste da significância dos resultados obtidos:

- H_0 : não há diferença no aprendizado de máquina durante o processo de regulação automatizado, quando se usa informações do histórico de atendimentos de um beneficiário;
- H_1 : existe diferença no aprendizado de máquina durante o processo de regulação automatizado, quando se usa informações do histórico de atendimentos de um beneficiário.

A partir dessas definições é possível calcular a zona de aceitação e a região rejeição para a hipótese nula e alternativa, respectivamente. Da forma como as hipóteses foram descritas, H_0 será aceita se as estimativas com uso de histórico fornecerem valores *Kappa* iguais a aqueles sem uso de histórico. Em contrapartida, se a estimativa com uso de histórico fornecer um valor *Kappa* diferente do valor obtido sem o uso de histórico, a hipótese alternativa será aceita. É importante lembrar que no teste de hipótese somente uma das hipóteses é aceita e a outra consequentemente é rejeitada, conforme definição apresentada abaixo:

$$\begin{cases} \text{Aceita } H_0 & : K_1 = K_2 \\ \text{Rejeita } H_1 & : K_1 \neq K_2 \end{cases}$$

Em que K_2 representa a estimativa do índice *Kappa* retornada pelo classificador com o uso de histórico. E o K_1 representa resultado da classificação sobre os dados sem uso do histórico.

Definida as hipóteses é necessário estabelecer um nível de significância, normalmente o valor estabelecido na maioria dos trabalhos é de 5%. Esse valor determina uma margem de rejeição para hipótese nula e por consequência delimita a zona de aceitação para a hipótese alternativa. O caminho comumente utilizado é provar a hipótese alternativa rejeitando a hipótese nula. Para o problema estudado, rejeitar H_0 é provar que os testes

com uso de histórico apresentam desempenho maior ou menor ao apresentado pelo teste sem histórico, tomando como base um determinado nível de significância.

Quando a hipótese alternativa é uma desigualdade tem-se um teste bilateral onde, para H_0 ser verdadeira a estimativa tem que ser próxima a Z. Porém se a estimativa for maior que $Z_{\frac{\alpha}{2}}$ ou menor que $-Z_{\frac{\alpha}{2}}$, H_0 é rejeitada. A Figura 16 apresenta uma ilustração exemplificando esse teste.

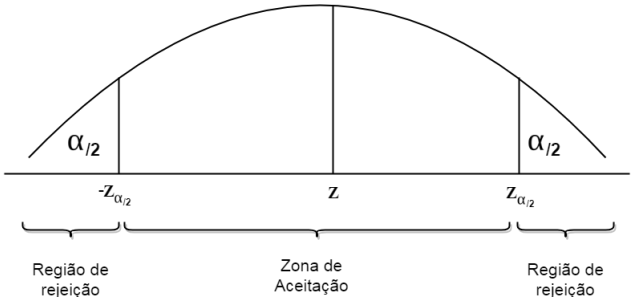

Figura 16 – Curva representativa da distribuição normal de probabilidade com a região de rejeição e zona de aceitação delimitados conforme definição das hipóteses apresentadas

3.6 Considerações Finais

Neste capítulo foram apresentados as principais referências teóricas utilizadas para se alcançar os objetivos deste trabalho. Inicialmente são detalhadas as principais técnicas representação de dados, seleção e transformação de atributos com objetivo de reduzir a dimensionalidade dos dados e melhorar a forma como os dados do histórico são dispostos na base de teste. Em seguida foi apresentado os métodos de avaliação escolhidos para verificar cada modelo criado pelas técnicas aplicadas. Esses métodos se baseiam nas principais métricas utilizadas para avaliação de alguns dos principais classificadores dos paradigmas simbólico, conexionista, baseado em exemplos estatístico. Também foi apresentada uma técnica de avaliação por meio de teste de hipótese baseado no índice Kappa dos classificadores testados.

No próximo capítulo será apresentado o sistema proposto, destacando os modelos obtidos a partir da aplicação das técnicas estudadas, as ferramentas utilizadas e os experimentos realizados.

4 Sistema Proposto

Neste capítulo é apresentado o sistema proposto pela pesquisa. Este sistema apresenta, como principal objetivo, uma modelagem baseada no histórico de solicitações, de maneira que a de aprendizagem em regulação médica apresente melhores resultados comparados a outras metodologias/abordagens que não usam histórico. A metodologia proposta, apresenta três etapas bem definidas: Pré-processamento, Predição e Avaliação de Performance. Cada uma dessas etapa estão detalhadas no fluxograma apresentado na Figura 17.

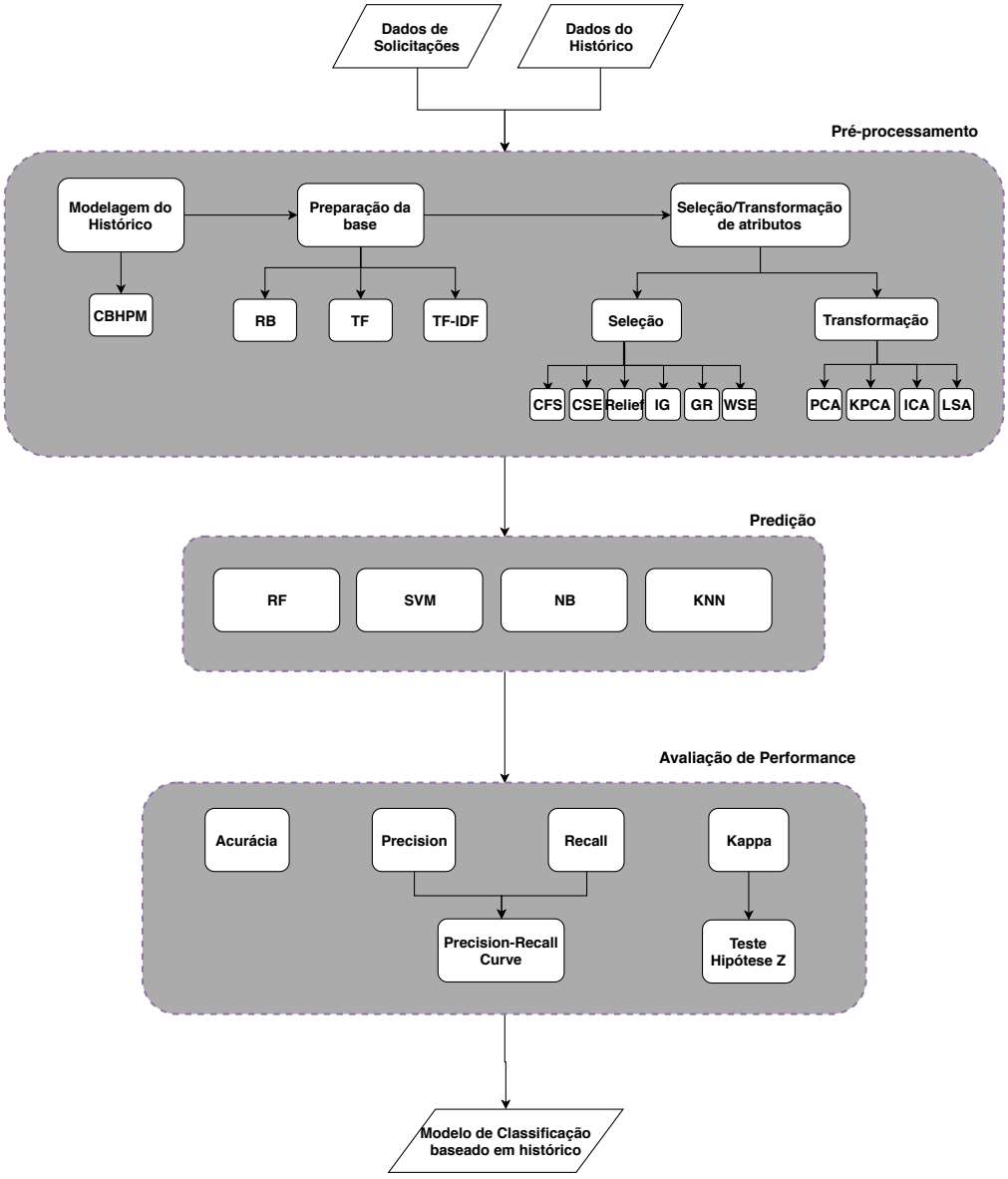

Figura 17 – Fluxograma da abordagem proposta

Para modelagem do histórico, preparação dos dados e classificação foram utilizadas

as ferramentas Weka (WITTEN et al., 2016) e a linguagem Python (PYTHON, 2019), bem como as bibliotecas **pandas**, **numpy**, **scikit-learn**.

4.1 Dados de Solicitações / Dados do Histórico

A abordagem proposta leva em consideração que inicialmente os dados da solicitação e do histórico estão divididas fisicamente em arquivos diferentes. Onde o primeiro apresenta os dados com as informações comuns de uma solicitação, enquanto o segundo apresenta informações sobre procedimentos/exames/tratamentos que foram autorizados para um determinado beneficiário. Essa informação pode estar contida em um arquivo só, porém nesses casos seria necessário uma etapa a mais do pré-processamento que seria a criação de uma estrutura ou arquivo que contenha essas informações do histórico de solicitações.

4.1.1 Base de dados

A base de dados utilizada foi fornecida por uma OPS privada brasileira (INFOWAY, 2018) com informações coletados no período de 01/01/2016 a 31/12/2016 referentes às internações de urgência. Esses dados são separados em dois arquivos, onde um refere-se às requisições realizadas pelos prestadores, juntamente com informações referentes à própria solicitação, e o outro com histórico de solicitações realizadas pelos beneficiários. O primeiro arquivo, nomeado de "dados_ops", apresenta 481 instâncias, deste total, somente 344 apresentam o valor de "Autorizado"preenchido. O segundo arquivo chamado de "dados_do_histórico"apresenta informações dos mesmos beneficiários presentes no primeiro documento, com 3593 instâncias, onde é feita uma relação entre os dois arquivos através das informações do guia de consulta e número do beneficiário. As Tabelas 19 e 20 apresentam os atributos presentes em cada um dos arquivos e seus respectivos tipos.

Tabela 19 – Atributos do arquivo dados_ops

Nº	**Atributos**	**Tipo**
1	Código beneficiário	Nominal
2	Tipo de beneficiário	Nominal
3	Idade beneficiário	Numérico
4	Sexo beneficiário	Nominal
5	Guia de consulta	Nominal
6	Especialidade	Numério
7	Tipo de tratamento	Nominal
8	Data de marcação	Data
9	CID	Nominal
10	**Autorizado (Classe)**	Nominal

Tabela 20 – Atributos do arquivo dados_do_historico

Nº	**Atributos**	**Tipo**
1	Código beneficiário	Nominal
2	CBHPM	Nominal
3	Quantidade	Numérico
4	Guia de consulta	Nominal
5	Data da consulta	Data
6	CID	Nominal

4.2 Pré-processamento

A etapa de pré-processamento apresenta três processos principais: modelagem do histórico, preparação da base e seleção/transformação de atributos.

4.2.1 Modelagem do Histórico

Após a coleta dos dados, é realizada a modelagem do histórico baseada na tabela de Classificação Brasileira Hierarquizada de Procedimentos Médicos (CBHPM). Nesta modelagem, cada elemento do CBHPM é uma categoria de procedimento/tratamento/exame que um beneficiário realizou ou não no passado. Cada uma dessas categorias são modeladas como atributos na base final, onde cada um apresenta a informação sobre realização ou não daquele procedimento em específico.

Como o objetivo é analisar os efeitos que a inserção do histórico traz para aprendizagem em regulação, foram definidas duas versões da base final: uma modelada com histórico (ALL) e a outra somente com as informações comuns da solicitação sem o histórico (NO_HIS).

Na versão NO_HIS foram usadas os dados do arquivo "dados_ops"juntamente com as informações de CBHPM do arquivo "dados_do_historico". Porém somente as informações referentes ao CBHPM solicitadas é que são armazenadas, nenhuma informação sobre solicitações passadas são consideradas. Alguns atributos foram eliminados da base final como o código do beneficiário, data de marcação e guia de consulta, pois poderiam afetar a geração do modelo de classificação. Alguns atributos foram convertidos de nominal para binário como: CBHPM, CID, Tipo de tratamento e Tipo de Beneficiário. No final a base NO_HIS apresentou o conjunto de atributos demonstrados na Tabela 21. Nessa tabela são apresentados a posição do atributo ou a quantidade de posições que o atributo ocupa após a transformação (representada pela coluna Nº). A transformação de nominal para binário gerou uma nova base com 394 atributos no total.

Na versão da base com histórico (ALL) são inseridas cada categoria da Tabela CBHPM (AMB, 2016) como um atributo, onde cada um desses atributos informa a quantidade de vezes que o procedimento/tratamento/exame foi realizado pelo beneficiário.

Tabela 21 – Atributos da base NO_HIS após transformação dos atributos nominais em binários

Nº	**Atributos**	**Tipo**
1	Tipo de beneficiário	{0,1}
2	Idade beneficiário	Numérico
3	Sexo Beneficiário	{0,1}
4	Especialidade	Numérico
5-7	Tipo de Tratamento	{0,1}
8-161	CID	{0,1}
162-393	CBHPM	{0,1}
394	Autorizado (Classe)	{S,N}

Para essa versão da base, o conjunto de atributos total pode ser visualizado na Tabela 22. Nessa tabela é possível observar que são os mesmos dados da Tabela 21 porém com o acréscimo da informação do Histórico_CBHPM, que representa todos os 4160 procedimentos presentes no CBHPM modelados como atributos da base. No final, a base com histórico (ALL) apresenta 4554 atributos no total.

Após as transformações, ambas as bases, NO_HIS e ALL, apresentaram no total 2067 instâncias, onde 2012 pertencem à classe **S** (Solicitação autorizada) e 55 à classe **N** (Solicitação não-autorizada).

Tabela 22 – Atributos da base ALL com todos os atributos CBHPM

Nº	**Atributos**	**Tipo**
1	Tipo de beneficiário	{0,1}
2	Idade beneficiário	Numérico
3	Sexo Beneficiário	{0,1}
4	Especialidade	Numérico
5-7	Tipo de Tratamento	{0,1}
8-161	CID	{0,1}
162-393	CBHPM	{0,1}
394-4553	Historico_CBHPM	Numérico
4554	Autorizado (Classe)	{S,N}

4.2.2 Preparação da base

Na etapa de preparação da base, as informações do histórico modelado são representados de três formas diferentes: representação binária (RB), *term-frequency* (TF) e *term-frequency inverse document frequency* (TF-IDF).

Ao final desta etapa foram geradas quatro bases: uma sem histórico (NO_HIS) e três com histórico (ALL), cada uma com uma representação de dados diferente: Binário (BIN_ALL), TF (TF_ALL) e TF-IDF (TF_IDF_ALL).

4.2.3 Seleção/Transformação de atributos

Para cada uma das bases transformadas com histórico (BIN_ALL, TF_ALL e TF_IDF_ALL) são aplicados algoritmos de Seleção e Transformação de Atributos, com o

objetivo principal de reduzir a dimensionalidade das informações do histórico e melhorar a performance dos classificadores na etapa de predição. Para a seleção de atributos foram utilizados: *correlation-feature subset eval* (CFS), *consistency subset eval* (CSE), Ganho de Informação (*Information Gain* - IG), razão de ganho (*Gain Ratio* - GR), Relief e *wrapper subset eval* (WSE). Para transformação, foram testados os algoritmos *principal component analysis* (PCA), Kenel PCA, *independent component analysis* (ICA) e *latent semantic analysis* (LSA).

4.3 Predição

Nesta etapa, cada uma das combinações de representação com técnicas de seleção/transformações de atributos são testadas por quatro classificadores de cada um dos paradigmas simbólico (*Random Forest*), estatístico (*Naive Bayes*), baseado em exemplos (*K-nearest neighbors*) e conexionista (Support Vector Machine). O objetivo desta etapa é avaliar quais representações e técnicas de seleção/transformação tiveram resultados positivos para aprendizagem sob diferentes perspectivas de classificação.

Os testes foram executados usando validação cruzada com 10-folds, nos classificadores RF (configurado com 500 árvores), SVM (configurado com função kernel linear), KNN (configurado com 5 vizinhos e algoritmo de cálculo da distância entre os vizinhos automático) e NB (versão que utiliza a distribuição Bernoulli).

4.4 Avaliação de Performance

Para avaliar a performance dos classificadores, são necessárias métricas de avaliação. Essas métricas, basicamente, medem os acertos e os erros da classificação de diferentes formas. Sob certos aspectos, algumas métricas são melhores para realização de uma análise específica. Por exemplo, para o problema de regulação, as bases utilizadas para aprendizagem apresentam grande desbalanceamento entre as classes. Nesses casos, a análise das métricas de precisão e *recall* são preferíveis pois avaliam o acerto de uma classe específica. O índice Kappa também é outra técnica muito utilizada para avaliar o quão bem um classificador se saiu considerando níveis de qualidade. Além disso, a partir dessa técnica é possível realizar um teste comparativo entre as performances dos classificadores (Teste de hipótese Z). O objetivo desse teste é avaliar se os resultados dos classificadores utilizando a modelagem do histórico são estatisticamente melhores que os resultados com a base sem o histórico, considerando um nível de significância específico.

4.5 Modelo de Classificação baseado em histórico

Ao final, após a análise dos resultados retornados pelos classificadores, é gerado um modelo de classificação para o processo de regulação automatizado com o uso de informações do histórico dos beneficiários.

4.6 Considerações Finais

O sistema proposto é composto por um conjunto de ferramentas e procedimentos computacionais, onde estão destacados todas as etapas, desde a modelagem do histórico e preparação da base de dados até a entrega de um modelo de classificação baseado em histórico de beneficiários. Nesse sistema, é possível inserir o histórico de beneficiários na base de testes (modelado como um conjunto de atributos do problema), e, a partir de técnicas de seleção e transformação de atributos, melhorar o processo de aprendizagem da Regulação de uma OPS. No próximo capítulo são apresentados com profundidade os testes e resultados obtidos com a aplicação desta proposta.

5 Resultados e Discussão

Neste capítulo são apresentados os resultados para as variadas transformações da base com o histórico: BIN_ALL, TF_ALL, TF_IDF_ALL, bem como para a base sem histórico NO_HIS. Onde são avaliados os resultados dos classificadores RF, SVM, KNN e NB, aplicados para cada uma das técnicas de Seleção (CFS, CSE, GR, IG e WSE) e Transformação de atributos (PCA, KPCA, ICA e LSA). Ao final são analisados quais combinações de técnicas obtiveram melhores resultados, sempre comparando com os resultados sem uso do histórico (NO_HIS). Além disso, é realizado ao final um teste de significância para avaliar se a melhora obtida foi estatisticamente significativa e quais as combinações de técnicas se saíram melhor.

5.1 Seleção de Atributos

Após a preparação das bases com histórico (BIN_ALL, TF_ALL e TF_IDF_ALL) são aplicadas técnicas de seleção de atributos com o objetivo de reduzir a dimensionalidade dos dados e melhorar a performance da classificação. Nesta etapa foram testados 6 algoritmos de seleção de atributos: CFS, CSE, Relief, IG, GR e WSE. Nas próximas seções são apresentados os resultados da classificação para cada uma dessas técnicas, onde serão discutidos os parâmetros escolhidos, os efeitos da classificação para cada técnica e quais representações de dados (BIN, TF ou TF-IDF) se saírem melhor de acordo com a análise das métricas de avaliação. É importante destacar também que cada um dos resultados obtidos são comparados com os resultados da classificação da base sem histórico (NO_HIS).

5.1.1 Correlation-based Feature Subset - CFS

Neste teste foi utilizada a implementação do algoritmo CFS presente no WEKA. Os algoritmos de geração de subconjuntos combinados com o CFS foram: *Greedy Stepwise* (GSW), *Best First* (BF) e o *Genetic Search* (GS). Para cada algoritmo de geração de subconjunto foram testadas estratégias *forward*, *backward* e *randomly*. Porém os resultados utilizando a estratégia *backward* não puderam ser obtidos devido a quantidade e tamanho dos subconjuntos mantidos em memória, o que ocasionou estouro da pilha de execução. Dessa forma, para todos os testes com seleção de atributos, as únicas estratégias testadas foram a *forward* e *randomly*.

Para o algoritmo GSW foi utilizado o *threshold* igual -1.79E308, o BF foi setado 5 iterações como critério de parada, caso não haja melhora da solução encontrada. O

GS foi configurado com probabilidade de *crossover* de 0.6, máximo de gerações igual a 20, probabilidade de mutação de 0,033 e tamanho da população igual a 20. As Tabelas 23, 24, 25 e 26, apresentam os resultados dos classificadores RF, SVM, KNN e NB. Em cada tabela são apresentados os resultados da classificação sem histórico (NO_HIS) e com o histórico, para cada representação (BIN_ALL, TF_ALL e TF-IDF_ALL). Cada representação de dados é combinada com cada uma das configurações do algoritmo CFS: CFS_GSW, CFS_BF, CFS_GS. Para cada avaliação é apresentada a quantidade de atributos do subconjunto final, Acurácia, Kappa, Precisão (para classe S e N) e Recall (para classe S e N).

Tabela 23 – Resultado do classificador RF para a técnica de seleção de atributos CFS

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.97	0.3255	0.98	0.41	0.99	0.29
BIN_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_IDF_ALL	4553	0.9894	0.7746	0.99	0.87	1.0	0.71
BIN_CFS_BF	9	0.986	0.7362	0.99	0.72	0.99	0.76
BIN_CFS_GSW	9	0.986	0.7362	0.99	0.72	0.99	0.76
BIN_CFS_GS	961	0.9855	0.7041	0.99	0.76	0.99	0.67
TF_CFS_BF	9	0.9869	0.7585	0.99	0.73	0.99	0.8
TF_CFS_GSW	9	0.9869	0.7585	0.99	0.73	0.99	0.8
TF_CFS_GS	1950	0.9903	0.8064	0.99	0.84	1.0	0.78
TF_IDF_CFS_BF	9	0.9869	0.7585	0.99	0.73	0.99	0.8
TF_IDF_CFS_GSW	9	0.9869	0.7585	0.99	0.73	0.99	0.8
TF_IDF_CFS_GS	1096	0.986	0.6877	0.99	0.82	1.0	0.6

A partir dos resultados apresentados na Tabela 23, é possível concluir que todas as combinações de técnicas apresentadas tiveram resultados superiores ao resultado NO_HIS (sem histórico) considerando todas as métricas de avaliação testadas (Acurácia, Kappa, Precisão e Recall). A combinação que apresentou o melhor resultado, considerando todas as métricas avaliadas, neste teste, foi TF_CFS_GS (Representação TF com CFS combinado com algoritmo de geração de subconjunto genetic search (GS)). Um fato interessante a ser evidenciado são os resultados obtidos BIN_ALL, TF_ALL e TF_IDF_ALL. A maioria dos resultados obtidos com CFS conseguiram reduzir significativamente a dimensionalidade dos dados, porém, não houve uma melhora de performance em relação aos resultados BIN_ALL, TF_ALL e TF_IDF_ALL (histórico com todos os atributos).

A Tabela 24 apresenta os resultados do classificador SVM para cada uma das configurações do algoritmo CFS. Neste teste, têm-se que todas as combinações apresentaram resultados superiores ao NO_HIS. O melhor resultado é o apresentado pelo histórico com todos atributos com representação binária (BIN_ALL). Para o classificador SVM, nenhuma combinação com CFS conseguiu obter resultados superiores às versões BIN_ALL, TF_ALL e TF_IDF_ALL da base com histórico.

Analisando a Tabela 25, é possível perceber que os resultados com histórico mostraram-se superiores que NO_HIS. Onde o melhor resultado encontrado foi TF_CFS_GS (Base representada com TF e algoritmo CFS com geração de subconjunto GS).

Tabela 24 – Resultado do classificador SVM para a técnica de seleção de atributos CFS

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9744	0.3181	0.98	0.54	0.99	0.24
BIN_ALL	4553	0.9898	0.7869	0.99	0.87	1.0	0.73
TF_ALL	4553	0.9894	0.7701	0.99	0.88	1.0	0.69
TF_IDF_ALL	4553	0.9884	0.7588	0.99	0.83	1.0	0.71
BIN_CFS_BF	9	0.986	0.7315	0.99	0.73	0.99	0.75
BIN_CFS_GSW	9	0.986	0.7315	0.99	0.73	0.99	0.75
BIN_CFS_GS	961	0.9797	0.5612	0.99	0.65	0.99	0.51
TF_CFS_BF	9	0.9787	0.4124	0.98	0.76	1.0	0.29
TF_CFS_GSW	9	0.9787	0.4124	0.98	0.76	1.0	0.29
TF_CFS_GS	1950	0.9874	0.7229	0.99	0.85	1.0	0.64
TF_IDF_CFS_BF	9	0.9797	0.513	0.98	0.7	1.0	0.42
TF_IDF_CFS_GSW	9	0.9797	0.513	0.98	0.7	1.0	0.42
TF_IDF_CFS_GS	1096	0.9845	0.6206	0.99	0.87	1.0	0.49

Tabela 25 – Resultado do classificador KNN para a técnica de seleção de atributos CFS

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.984	0.6447	0.99	0.78	1.0	0.56
BIN_ALL	4553	0.9884	0.7633	0.99	0.82	1.0	0.73
TF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
TF_IDF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
BIN_CFS_BF	9	0.9869	0.741	0.99	0.77	0.99	0.73
BIN_CFS_GSW	9	0.9869	0.741	0.99	0.77	0.99	0.73
BIN_CFS_GS	961	0.9884	0.7492	0.99	0.86	1.0	0.67
TF_CFS_BF	9	0.9865	0.7385	0.99	0.75	0.99	0.75
TF_CFS_GSW	9	0.9865	0.7385	0.99	0.75	0.99	0.75
TF_CFS_GS	1950	0.9913	0.8119	0.99	0.93	1.0	0.73
TF_IDF_CFS_BF	9	0.9855	0.6985	0.99	0.77	0.99	0.65
TF_IDF_CFS_GSW	9	0.9855	0.6985	0.99	0.77	0.99	0.65
TF_IDF_CFS_GS	1096	0.9879	0.7511	0.99	0.81	1.0	0.71

Tabela 26 – Resultado do classificador NB para a técnica de seleção de atributos CFS

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_CFS_BF	9	0.9845	0.6446	0.99	0.81	1.0	0.55
BIN_CFS_GSW	9	0.9845	0.6446	0.99	0.81	1.0	0.55
BIN_CFS_GS	961	0.9734	0.0	0.97	0.0	1.0	0.0
TF_CFS_BF	9	0.9845	0.6446	0.99	0.81	1.0	0.55
TF_CFS_GSW	9	0.9845	0.6446	0.99	0.81	1.0	0.55
TF_CFS_GS	1950	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_CFS_BF	9	0.9845	0.6446	0.99	0.81	1.0	0.55
TF_IDF_CFS_GSW	9	0.9845	0.6446	0.99	0.81	1.0	0.55
TF_IDF_CFS_GS	1096	0.9734	0.0	0.97	0.0	1.0	0.0

Os resultados obtidos com NB (Tabela 26) foram os mais modestos, comparados aos outros classificadores. As versões NO_HIS, BIN_ALL, TF_ALL, TF_IDF_ALL, BIN_CFS_GS, TF_CF_GS e TF_IDF_CFS_GS, não tiveram sucesso para classificar os registros da classe N. Devido ao desbalanceamento da base, a métrica Acurácia permanece alta, porém o índice kappa é 0, e a taxa de Precisão e Recall para a classe N também. O que evidencia que o classificador não acertou nenhuma predição para a classe N. Algo interessante a se notar é que as versões que obtiveram os piores resultados foram as que apresentaram a maior quantidade de atributos. Todos as outras combinações, exceto essas apontadas anteriormente, apresentaram os mesmos resultados em todas as métricas avaliadas (0.9845 de acurácia, 0.6446 de kappa, Precisão S de 0.99, Precisão N de 0.81,

Recall S igual a 1 e Recall N igual a 0.55).

5.1.2 Consistency Subset Eval - CSE

Assim como o algoritmo CFS, o algoritmo CSE foi combinado com os algoritmos de geração de subconjunto GSW, GS e BF com as mesmas configurações. Os resultados para cada classificador é apresentado nas Tabelas 27, 28, 29 e 30. Onde os melhores resultados foram obtidos com as combinações TF_CSE_GSW (Base com representação TF combinada com o algoritmo CSE com geração de subconjunto GSW) e TF_IDF_CSE_GSW (Representação TF-IDF combinada com algoritmo CSE com geração de subconjunto GSW).

Tabela 27 – Resultado do classificador RF para a técnica de seleção de atributos CSE

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.97	0.3255	0.98	0.41	0.99	0.29
BIN_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_IDF_ALL	4553	0.9894	0.7746	0.99	0.87	1.0	0.71
BIN_CSE_BF	11	0.9894	0.7945	0.99	0.8	0.99	0.8
BIN_CSE_GSW	9	0.9903	0.8064	0.99	0.84	1.0	0.78
BIN_CSE_GS	2129	0.9923	0.8451	1.0	0.88	1.0	0.82
TF_CSE_BF	11	0.9903	0.8165	1.0	0.81	0.99	0.84
TF_CSE_GSW	9	0.9913	0.8289	1.0	0.85	1.0	0.82
TF_CSE_GS	2230	0.9908	0.8143	0.99	0.86	1.0	0.78
TF_IDF_CSE_BF	11	0.9903	0.8165	1.0	0.81	0.99	0.84
TF_IDF_CSE_GSW	9	0.9913	0.8289	1.0	0.85	1.0	0.82
TF_IDF_CSE_GS	2230	0.9913	0.8225	0.99	0.88	1.0	0.78

A Tabela 27 apresenta os resultados do classificador RF para diferentes combinações do algoritmo CSE. É possível perceber que os resultados obtidos foram superiores às versões NO_HIS, BIN_ALL, TF_ALL e TF_IDF_ALL. Podendo-se concluir que foi bem sucedida em reduzir a dimensionalidade e melhorar a performance da classificação.

Tabela 28 – Resultado do classificador SVM para a técnica de seleção de atributos CSE

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9744	0.3181	0.98	0.54	0.99	0.24
BIN_ALL	4553	0.9898	0.7869	0.99	0.87	1.0	0.73
TF_ALL	4553	0.9894	0.7701	0.99	0.88	1.0	0.69
TF_IDF_ALL	4553	0.9884	0.7588	0.99	0.83	1.0	0.71
BIN_CSE_BF	11	0.9869	0.741	0.99	0.77	0.99	0.73
BIN_CSE_GSW	9	0.9836	0.6058	0.99	0.82	1.0	0.49
BIN_CSE_GS	2129	0.9869	0.741	0.99	0.77	0.99	0.73
TF_CSE_BF	11	0.9787	0.3972	0.98	0.79	1.0	0.27
TF_CSE_GSW	9	0.9782	0.3585	0.98	0.81	1.0	0.24
TF_CSE_GS	2230	0.9894	0.7655	0.99	0.9	1.0	0.67
TF_IDF_CSE_BF	11	0.9802	0.485	0.98	0.77	1.0	0.36
TF_IDF_CSE_GSW	9	0.9787	0.3972	0.98	0.79	1.0	0.27
TF_IDF_CSE_GS	2230	0.9898	0.7869	0.99	0.87	1.0	0.73

A Tabela 28 apresenta os resultados para o classificador SVM para o algoritmo CSE. É possível perceber que os resultados retornados pelo classificador se assemelham muito aos resultados apresentados pela SVM para a técnica CFS (Tabela 24). O melhor resultado

obtido foi usando todos os atributos do histórico com representação binária (BIN_ALL). É importante destacar que todas as combinações do CSE obtiveram resultados superiores ao NO_HIS, com as combinações com algoritmo de geração de subconjunto GS apresentando os melhores resultados.

Tabela 29 – Resultado do classificador KNN para a técnica de seleção de atributos CSE

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.984	0.6447	0.99	0.78	1.0	0.56
BIN_ALL	4553	0.9884	0.7633	0.99	0.82	1.0	0.73
TF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
TF_IDF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
BIN_CSE_BF	11	0.9898	0.7869	0.99	0.87	1.0	0.73
BIN_CSE_GSW	9	0.9918	0.8206	0.99	0.95	1.0	0.73
BIN_CSE_GS	2129	0.9874	0.7483	0.99	0.78	0.99	0.73
TF_CSE_BF	11	0.9908	0.8034	0.99	0.91	1.0	0.73
TF_CSE_GSW	9	0.9918	0.8206	0.99	0.95	1.0	0.73
TF_CSE_GS	2230	0.9903	0.7951	0.99	0.89	1.0	0.73
TF_IDF_CSE_BF	11	0.9908	0.8034	0.99	0.91	1.0	0.73
TF_IDF_CSE_GSW	9	0.9918	0.8206	0.99	0.95	1.0	0.73
TF_IDF_CSE_GS	2230	0.9903	0.7951	0.99	0.89	1.0	0.73

Os resultados do algoritmo KNN para o CSE (Tabela 29) apontam o algoritmo de geração de subconjunto GSW combinado com CSE obtiveram os melhores resultados em todas as representações (BIN, TF e TF_IDF)

Tabela 30 – Resultado do classificador NB para a técnica de seleção de atributos CSE

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_ALL_ATTR	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_ALL_ATTR	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_ALL_ATTR	4553	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_CSE_BF	11	0.9845	0.629	0.99	0.85	1.0	0.51
BIN_CSE_GSW	9	0.9855	0.6596	0.99	0.86	1.0	0.55
BIN_CSE_GS	2129	0.9734	0.0	0.97	0.0	1.0	0.0
TF_CSE_BF	11	0.9845	0.629	0.99	0.85	1.0	0.51
TF_CSE_GSW	9	0.9855	0.6596	0.99	0.86	1.0	0.55
TF_CSE_GS	2230	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_CSE_BF	11	0.9845	0.629	0.99	0.85	1.0	0.51
TF_IDF_CSE_GSW	9	0.9855	0.6596	0.99	0.86	1.0	0.55
TF_IDF_CSE_GS	2230	0.9734	0.0	0.97	0.0	1.0	0.0

O resultados do algoritmo NB para a técnica CSE (Tabela 30) são muito parecidos com os resultados anteriores do classificador para a técnica CFS, porém com uma melhora. A partir desses resultados é possível perceber que o NB tem dificuldade para classificar a base com muitos atributos. Além disso, essa dificuldade se reflete na ineficiência de classificar corretamente os registros da classe minoritária N.

5.1.3 Gain Ratio (Razão de Ganho) - GR

A implementação do algoritmo GR presente no WEKA, permite apenas o uso de algoritmos baseados em *Ranker Search*(RS) para geração de subconjunto. Por isso, somente esta opção de geração de subconjunto foi testada. Os resultados dos classificadores para esta técnica podem ser vistos nas Tabelas 31, 32, 33 e 34.

Tabela 31 – Resultado do classificador RF para a técnica de seleção de atributos GR

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.97	0.3255	0.98	0.41	0.99	0.29
BIN_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_IDF_ALL	4553	0.9894	0.7746	0.99	0.87	1.0	0.71
BIN_GR_RS	62	0.9927	0.8611	1.0	0.86	1.0	0.87
TF_GR_RS	38	0.9927	0.8658	1.0	0.83	1.0	0.91
TF_IDF_GR_RS	38	0.9927	0.8658	1.0	0.83	1.0	0.91

Os resultados obtidos apresentados pela Tabela 31, apontam que todas as combinações das técnicas GR com as representações BIN, TF e TF_IDF, apresentaram os melhores resultados com a representação BIN levemente inferior às outras.

Tabela 32 – Resultado do classificador SVM para a técnica de seleção de atributos GR

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9744	0.3181	0.98	0.54	0.99	0.24
BIN_ALL	4553	0.9898	0.7869	0.99	0.87	1.0	0.73
TF_ALL	4553	0.9894	0.7701	0.99	0.88	1.0	0.69
TF_IDF_ALL	4553	0.9884	0.7588	0.99	0.83	1.0	0.71
BIN_GR_RS	62	0.9879	0.7602	0.99	0.79	0.99	0.75
TF_GR_RS	38	0.9806	0.4915	0.98	0.8	1.0	0.36
TF_IDF_GR_RS	38	0.9806	0.5149	0.98	0.76	1.0	0.4

A Tabela 32 apresenta os resultados do classificador SVM para a técnica GR. Apesar dos ótimos resultados obtidos com o classificador RF, o mesmo não ocorre com o classificador SVM. Apesar da técnica GR apresentar resultados superiores em comparação à versão NO_HIS, o mesmo não pode ser dito quando comparado às versões BIN_ALL, TF_ALL e TF_IDF_ALL, que foram levemente inferiores.

Tabela 33 – Resultado do classificador KNN para a técnica de seleção de atributos GR

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.984	0.6447	0.99	0.78	1.0	0.56
BIN_ALL	4553	0.9884	0.7633	0.99	0.82	1.0	0.73
TF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
TF_IDF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
BIN_GR_RS	62	0.9889	0.771	0.99	0.83	1.0	0.73
TF_GR_RS	38	0.9903	0.7951	0.99	0.89	1.0	0.73
TF_IDF_GR_RS	38	0.9903	0.7951	0.99	0.89	1.0	0.73

Os resultados do classificador KNN (Tabela 33 apontam que a representação Binária (BIN) obteve resultados inferiores às outras representações. Também apontam que os melhores resultados foram aqueles que apresentaram representação TF e TF_IDF com todos os atributos do histórico (ALL).

O classificador NB (Tabela 34), de novo, apresentou resultados muito inferiores comparados aos outros classificadores. Apesar dos resultados com uso do algoritmo GR terem trazido uma melhora, esta foi bastante inferior se comparada aos resultados obtidos pelas técnicas CFS e CSE para o mesmo classificador.

Tabela 34 – Resultado do classificador NB para a técnica de seleção de atributos GR

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_ALL_ATTR	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_ALL_ATTR	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_ALL_ATTR	4553	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_GR_RS	62	0.8399	0.2128	1.0	0.14	0.84	1.0
TF_GR_RS	38	0.8283	0.199	1.0	0.13	0.82	1.0
TF_IDF_GR_RS	38	0.8345	0.1992	1.0	0.13	0.83	0.96

5.1.4 Information Gain (Ganho de Informação) - IG

Assim como nos testes realizados com GR, a técnica IG só pôde ser combinada com o algoritmo *Ranker Search* (RS). As Tabelas 35, 36, 37 e 38 apresentam os resultado dos classificadores para o algoritmo IG. Os resultados são idênticos aos das Tabelas 31, 32, 33 e 34 pois o conjunto de atributos selecionados por ambos algoritmos foram os mesmos.

Tabela 35 – Resultado do classificador RF para a técnica de seleção de atributos IG

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.97	0.3255	0.98	0.41	0.99	0.29
BIN_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_IDF_ALL	4553	0.9894	0.7746	0.99	0.87	1.0	0.71
BIN_IG_RS	62	0.9927	0.8611	1.0	0.86	1.0	0.87
TF_IG_RS	38	0.9927	0.8658	1.0	0.83	1.0	0.91
TF_IDF_IG_RS	38	0.9927	0.8658	1.0	0.83	1.0	0.91

Tabela 36 – Resultado do classificador SVM para a técnica de seleção de atributos IG

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9744	0.3181	0.98	0.54	0.99	0.24
BIN_ALL	4553	0.9898	0.7869	0.99	0.87	1.0	0.73
TF_ALL	4553	0.9894	0.7701	0.99	0.88	1.0	0.69
TF_IDF_ALL	4553	0.9884	0.7588	0.99	0.83	1.0	0.71
BIN_IG_RS	62	0.9879	0.7602	0.99	0.79	0.99	0.75
TF_IG_RS	38	0.9806	0.4915	0.98	0.8	1.0	0.36
TF_IDF_IG_RS	38	0.9806	0.5149	0.98	0.76	1.0	0.4

Tabela 37 – Resultado do classificador KNN para a técnica de seleção de atributos IG

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.984	0.6447	0.99	0.78	1.0	0.56
BIN_ALL	4553	0.9884	0.7633	0.99	0.82	1.0	0.73
TF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
TF_IDF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
BIN_IG_RS	62	0.9889	0.771	0.99	0.83	1.0	0.73
TF_IG_RS	38	0.9903	0.7951	0.99	0.89	1.0	0.73
TF_IDF_IG_RS	38	0.9903	0.7951	0.99	0.89	1.0	0.73

5.1.5 Relief

Para o algoritmo Relief foram testadas uma vizinhança igual a 50 para a representação Binária (BIN_RELIEF_RS) e vizinhança igual a 10 para as representações TF

Tabela 38 – Resultado do classificador NB para a técnica de seleção de atributos IG

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_IG_RS	62	0.8399	0.2128	1.0	0.14	0.84	1.0
TF_IG_RS	38	0.8283	0.199	1.0	0.13	0.82	1.0
TF_IDF_IG_RS	38	0.8345	0.1992	1.0	0.13	0.83	0.96

e TF-IDF (TF_RELIEF_RS e TF_IDF_RS). Assim como os algoritmos GR e IG, só foi possível realizar combinação com o algoritmo RS. Os resultados para o algoritmo RF (Tabela 39) apontam que a representação TF_IDF apresentou os melhores resultados, onde a melhor combinação foi a TF_IDF_RELIEF_RS.

Tabela 39 – Resultado do classificador RF para a técnica de seleção de atributos Relief

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.97	0.3255	0.98	0.41	0.99	0.29
BIN_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_IDF_ALL	4553	0.9894	0.7746	0.99	0.87	1.0	0.71
BIN_RELIEF_RS	169	0.9889	0.771	0.99	0.83	1.0	0.73
TF_RELIEF_RS	247	0.9884	0.7633	0.99	0.82	1.0	0.73
TF_IDF_RELIEF_RS	244	0.9894	0.783	0.99	0.84	1.0	0.75

Para o algoritmo SVM (Tabela 40), o melhor resultado obtido foi a combinação da representação BIN com todos os atributos do histórico (BIN_ALL).

Tabela 40 – Resultado do classificador SVM para a técnica de seleção de atributos Relief

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9744	0.3181	0.98	0.54	0.99	0.24
BIN_ALL	4553	0.9898	0.7869	0.99	0.87	1.0	0.73
TF_ALL	4553	0.9894	0.7701	0.99	0.88	1.0	0.69
TF_IDF_ALL	4553	0.9884	0.7588	0.99	0.83	1.0	0.71
BIN_RELIEF_RS	169	0.9884	0.7492	0.99	0.86	1.0	0.67
TF_RELIEF_RS	247	0.9879	0.7414	0.99	0.84	1.0	0.67
TF_IDF_RELIEF_RS	244	0.9879	0.7463	0.99	0.83	1.0	0.69

A Tabela 41 apresenta os resultados do classificador KNN, onde os melhores resultados foram aqueles que apresentaram representações em TF e TF-IDF para o histórico com todos os atributo (ALL). Com relação os resultados combinados com o algoritmo Relief, têm-se que a representação TF obteve os resultados inferiores em relação às outras.

Os resultados para o algoritmo NB pode ser visualizado na Tabela 42. Assim como os resultados anteriores, os resultados permanecem inferiores em comparação aos outros classificadores. A combinação de técnicas que apresentou o melhor resultado para este classificador foi a TF_IDF_RELIEF_RS.

Tabela 41 – Resultado do classificador KNN para a técnica de seleção de atributos Relief

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.984	0.6447	0.99	0.78	1.0	0.56
BIN_ALL	4553	0.9884	0.7633	0.99	0.82	1.0	0.73
TF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
TF_IDF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
BIN_RELIEF_RS	169	0.9903	0.7951	0.99	0.89	1.0	0.73
TF_RELIEF_RS	247	0.9894	0.7789	0.99	0.85	1.0	0.73
TF_IDF_RELIEF_RS	244	0.9903	0.7951	0.99	0.89	1.0	0.73

Tabela 42 – Resultado do classificador NB para a técnica de seleção de atributos Relief

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_RELIEF_RS	169	0.8384	0.211	1.0	0.14	0.83	1.0
TF_RELIEF_RS	247	0.8457	0.2129	1.0	0.14	0.84	0.96
TF_IDF_RELIEF_RS	244	0.8597	0.2286	1.0	0.15	0.86	0.95

5.1.6 Wrapper Subset Eval - WSE

A implementação do WSE utilizada foi também a presente no Weka. Como o WSE é uma técnica baseada em *wrapper*, então a avaliação do subconjunto gerado se dá pela utilização de um algoritmo de classificação. Nos testes apresentados foi escolhido o RF com configuração de 100 árvores, tendo a métrica acurácia como medida de avaliação, e testes de estimação da acurácia do subconjunto feitos utilizando 5-folds. O WSE foi combinado com os algoritmos BF, GSW e GS e não foram testadas outras combinações pelas mesmas razões apresentadas nos resultados das técnicas CFS e CSE.

A Tabela 43 apresenta os resultados da classificação do RF para as técnicas combinadas com o WSE. A partir das informações apresentadas, é possível perceber que os melhores resultados foram aqueles que apresentaram as configurações: BIN_WSE_BF, BIN_WSE_GS, TF_WSE_GS e TF_IDF_BF. É interessante notar que a algumas combinações retornaram apenas 1 atributo e que, apesar da classificação não ter sido boa, ela ainda foi superior à classificação da base sem histórico (NO_HIS).

Tabela 43 – Resultado do classificador RF para a técnica de seleção de atributos WSE

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.97	0.3255	0.98	0.41	0.99	0.29
BIN_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_IDF_ALL	4553	0.9894	0.7746	0.99	0.87	1.0	0.71
BIN_WSE_BF	11	0.9903	0.8064	0.99	0.84	1.0	0.78
BIN_WSE_GSW	1	0.9797	0.376	0.98	1.0	1.0	0.24
BIN_WSE_GS	2181	0.9918	0.8308	0.99	0.9	1.0	0.78
TF_WSE_BF	1	0.9797	0.376	0.98	1.0	1.0	0.24
TF_WSE_GSW	1	0.9797	0.376	0.98	1.0	1.0	0.24
TF_WSE_GS	2290	0.9918	0.8339	0.99	0.88	1.0	0.8
TF_IDF_WSE_BF	7	0.9918	0.8369	1.0	0.87	1.0	0.82
TF_IDF_WSE_GSW	1	0.9797	0.376	0.98	1.0	1.0	0.24
TF_IDF_WSE_GS	2170	0.9879	0.7463	0.99	0.83	1.0	0.69

Os resultados para o classificador SVM (Tabela 44) apontam que a melhor combinação de técnicas foi utilizando a representação binária com todos os atributos do histórico (BIN_ALL). Os resultados combinando a técnica WSE foram superiores aos resultados sem histórico (NO_HIS).

Tabela 44 – Resultado do classificador SVM para a técnica de seleção de atributos WSE

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9744	0.3181	0.98	0.54	0.99	0.24
BIN_ALL	4553	0.9898	0.7869	0.99	0.87	1.0	0.73
TF_ALL	4553	0.9894	0.7701	0.99	0.88	1.0	0.69
TF_IDF_ALL	4553	0.9884	0.7588	0.99	0.83	1.0	0.71
BIN_WSE_BF	11	0.984	0.6131	0.99	0.84	1.0	0.49
BIN_WSE_GSW	1	0.9797	0.376	0.98	1.0	1.0	0.24
BIN_WSE_GS	2181	0.9879	0.7557	0.99	0.8	1.0	0.73
TF_WSE_BF	1	0.9797	0.376	0.98	1.0	1.0	0.24
TF_WSE_GSW	1	0.9797	0.376	0.98	1.0	1.0	0.24
TF_WSE_GS	2290	0.9811	0.4982	0.98	0.83	1.0	0.36
TF_IDF_WSE_BF	7	0.9792	0.4031	0.98	0.83	1.0	0.27
TF_IDF_WSE_GSW	1	0.9797	0.376	0.98	1.0	1.0	0.24
TF_IDF_WSE_GS	2170	0.9836	0.6058	0.99	0.82	1.0	0.49

Tabela 45 – Resultado do classificador KNN para a técnica de seleção de atributos WSE

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.984	0.6447	0.99	0.78	1.0	0.56
BIN_ALL	4553	0.9884	0.7633	0.99	0.82	1.0	0.73
TF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
TF_IDF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
BIN_WSE_BF	11	0.9918	0.8206	0.99	0.95	1.0	0.73
BIN_WSE_GSW	1	0.9797	0.376	0.98	1.0	1.0	0.24
BIN_WSE_GS	2181	0.9884	0.7633	0.99	0.82	1.0	0.73
TF_WSE_BF	1	0.9797	0.376	0.98	1.0	1.0	0.24
TF_WSE_GSW	1	0.9797	0.376	0.98	1.0	1.0	0.24
TF_WSE_GS	2290	0.9913	0.8119	0.99	0.93	1.0	0.73
TF_IDF_WSE_BF	7	0.9918	0.8206	0.99	0.95	1.0	0.73
TF_IDF_WSE_GSW	1	0.9797	0.376	0.98	1.0	1.0	0.24
TF_IDF_WSE_GS	2170	0.9908	0.8034	0.99	0.91	1.0	0.73

Tabela 46 – Resultado do classificador NB para a técnica de seleção de atributos WSE

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_WSE_BF	11	0.9836	0.5877	0.99	0.86	1.0	0.45
BIN_WSE_GSW	1	0.9797	0.376	0.98	1.0	1.0	0.24
BIN_WSE_GS	2181	0.9734	0.0	0.97	0.0	1.0	0.0
TF_WSE_BF	1	0.9797	0.376	0.98	1.0	1.0	0.24
TF_WSE_GSW	1	0.9797	0.376	0.98	1.0	1.0	0.24
TF_WSE_GS	2290	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_WSE_BF	7	0.9836	0.5877	0.99	0.86	1.0	0.45
TF_IDF_WSE_GSW	1	0.9797	0.376	0.98	1.0	1.0	0.24
TF_IDF_WSE_GS	2170	0.9734	0.0	0.97	0.0	1.0	0.0

Para o classificador KNN (Tabela 45), as combinações BIN_WSE_BF e TF_IDF_WSE_BF obtiveram os melhores resultados. Para o classificador NB (Tabela 46), essas também as melhores combinações de classificação. Isso demonstra que o subconjunto gerado pelo algoritmo BF, combinado com as representações BIN e TF_IDF contém atributos que

ajudaram a melhorar a performance do classificador e reduzir a dimensionalidade dos dados.

5.1.7 Melhores resultados seleção de atributos

Aqui são apresentados um resumo com os principais resultados com o uso da seleção de atributos. A Tabela 47 evidencia, para cada classificador, as técnicas de seleção de atributos aplicadas às bases BIN_ALL, TF_ALL e TF_IDF_ALL que obtiveram os melhores resultados. Para cada resultado, são apresentados o Número de Atributos, gerados após a seleção de atributos, além das métricas Acurácia, Kappa, Precisão e Recall. O melhor resultado, considerando índice Kappa e Recall da classe N como métricas principais, foi o obtido com RF e seleção de atributos baseado em GR e IG e representação dos dados em TF e TF-IDF. Cada uma dessas técnicas retornaram um subconjunto de 38 atributos, obtendo Kappa de 0.8658 e Recall da classe N de 0,91.

Tabela 47 – Tabela com os melhores resultados de para cada classificador usando seleção de atributos

Classificador	Representação dos dados	N. Atributos	Acurária	Kappa	Precisão S	Precisão N	Recall S	Recall N
RF	TF_GR_RS TF_IDF_GR_RS TF_IG_RS TF_IDF_IG_RS	38 38 38 38	0.9927	0.8658	1.0	0.83	1.0	0.91
SVM	TF_IDF_CSE_GS	2230	0.9898	0.7869	0.99	0.87	1.0	0.73
KNN	BIN_CSE_GSW TF_CSE_GSW TF_IDF_CSE_GSW BIN_WSE_BF TF_IDF_WSE_BF	9 9 9 11 7	0.9918	0.8206	0.99	0.95	1.0	0.73
NB	BIN_CSE_GSW TF_CSE_GSW TF_IDF_CSE_GSW	9 9 9	0.9855	0.6596	0.99	0.86	1.0	0.55

Com relação ao resultado do SVM, apesar de menos expressivos comparados ao RF e KNN, apresentou como destaque o CSE aplicado à representação TF_IDF e algoritmo de geração de subconjunto GS. Com essa técnica foram selecionados 2230 atributos com 0.7869 de Kappa e 0.73 de Recall N.

Os melhores resultados para o KNN apontam que os algoritmo CSE e WSE obtiveram os melhores subconjuntos de atributos combinados com o GSW e o BF. Este foi o segundo melhor resultado geral de classificação. Já com relação ao NB, têm-se que o algoritmo CSE apresentou os melhores resultados para cada tipo de representação. O algoritmo NB não apresentou bons resultados de modo geral, é possível que isso esteja relacionado a dimensionalidade da base, pois os melhores resultados para este classificador foram aqueles com menor quantidade de atributos. Dessa forma, pode-se concluir que o CSE combinado com o GSW foi algoritmo de seleção de atributos que gerou os melhores subconjuntos para generalizar o comportamento da base. Com relação às representações, têm-se que a representação TF_IDF esteve em um número maior de resultados, o que leva a concluir que foi o melhor tipo de representação para a base.

A Tabela 48 apresenta todos os atributos retornados pelas técnicas que apresentaram os melhores resultados para os classificadores RF, KNN e NB. Para cada atributo é marcado se ele esteve presente ou não nos resultados de cada classificador. Além disso, apresenta a informação de se o atributo selecionado é referente à solicitação ou ao histórico. A técnica com melhor resultado para o classificador SVM resultou em 2230 atributos e, devido a extensão, não foi incluída na tabela. A melhor técnica para o RF selecionou 38 atributos, as demais técnicas para os outros classificadores selecionaram em torno de 7 a 11 atributos. Alguns atributos são compartilhados por todas as técnicas que são: CID=M54.2 (Cervicalgia), CID=J03.9 (Amigdalite aguda não especificada), CID=N39.0 (Infecção do trato urinário de localização não especificada), CID=O34.2 (Assistência prestada à mãe por cicatriz uterina devida a uma cirurgia anterior), CID=N13.3 (Outras hidronefroses e as não especificadas), CID=J20.9 (Bronquite aguda não especificada), Visita hospitalar (paciente internado), Idade do beneficiário e Tipo do beneficiário=2. Como esses atributos estavam presentes nos melhores resultados de cada classificador, o que se pode concluir é que eles são de extrema importância para o contexto da Regulação Médica considerando a base analisada. Os atributos presentes nos resultados do KNN são quase idênticos aos presentes no NB, exceto pelos atributos CBHPM=41001117 (Pelve_ou_bacia), Lipase, Coagulograma_(TS_TC_prova_do_laco_retracao_do_coagulo_contagem_de_plaquetas_tempo_de_protombina_tempo_de_tromboplastina_parcial_ativado). Uma outra informação significativa que se pode tirar dessa tabela é o fato da maioria dos atributos selecionados serem referentes ao histórico. Isso demonstra o quanto o histórico dos beneficiários agrega ao aprendizado em Regulação.

Tabela 48 – Atributos selecionados das técnicas com melhor avaliação do RF, KNN e NB

N. Atributo	Atributos	M. RF	M. KNN	M. NB	Tipo de Atributo
1	Potassio	X	-	-	histórico
2	Atendimento_medico_do_intensivista_em_UTI_geral_ou_pediatrica_(plantao_de_12_horas_-_por_paciente)	X	-	-	histórico
3	Fosfatase_alcalina	X	-	-	histórico
4	Transaminase_piruvica_(amino_transferase_de_alanina)	X	-	-	histórico
5	CID=M54.2	X	X	X	solicitação
6	Transaminase_oxalacetica_(amino_transferase_aspartato)	X	-	-	histórico
7	CID=J03.9	X	X	X	solicitação
8	Autorizado	X	X	X	classe
9	CID=I64	X	-	-	solicitação
10	Albumina	X	-	-	histórico
11	CID=N39.0	X	X	X	solicitação
12	CBHPM=41001117	-	X	-	solicitação
13	CID=N20.2	X	-	-	solicitação
14	Tipo_de_Tratamento=2	X	-	-	solicitação
15	ECG_convencional_de_ate_12_derivacoes	X	-	-	histórico
16	Proteina_C_reativa_qualitativa	X	-	-	histórico
17	CID=D57.0	X	-	-	solicitação
18	acido_lactico_(lactato)	X	-	-	histórico
19	Cloro	X	-	-	histórico
20	Especialidade	X	-	-	solicitação
21	CID=O34.2	X	X	X	solicitação
22	Glicose	X	-	-	histórico
23	CID=N13.3	X	X	X	solicitação
24	Calcio_ionico	X	-	-	histórico
25	Hemograma_com_contagem_de_plaquetas_ou_fracoes_(eritrograma_leucograma_plaquetas)	X	-	-	histórico
26	Sodio	X	-	-	histórico
27	Lipase	-	X	-	histórico
28	CID=J20.9	X	X	X	solicitação
29	Magnesio	X	-	-	histórico
30	Tipo_de_Tratamento=1	X	-	-	solicitação
31	Fosforo	X	-	-	histórico
32	Atendimento_do_intensivista_diarista_(por_dia_e_por_paciente)	X	-	-	histórico
33	Visita_hospitalar_(paciente_internado)	X	X	X	histórico
34	Bilirrubinas_(direta_indireta_e_total)	X	-	-	histórico
35	Calcio	X	-	-	histórico
36	Creatinina	X	-	-	histórico
37	Gasometria_(pH_pCO2_SA_O2_excesso_base)	X	-	-	histórico
38	Gama-glutamil_transferase	X	-	-	histórico
39	Ureia	X	-	-	histórico
40	Coagulograma_(TS_TC_prova_do_laco_retracao_do_coagulo_contagem_de_plaquetas_tempo_de_protombina_tempo_de_tromboplastina_parcial_ativado)	-	X	-	histórico
41	Amilase	-	X	-	histórico
42	Idade_Beneficiario	X	X	X	solicitação
43	Tipo_Beneficiario=2	X	X	X	solicitação

As Figuras 18, 19, 20 e 21 apresentam as curvas Precision-Recall para a classe S e N de cada um dos classificadores testados. Cada gráfico apresenta também, além do melhor resultado de cada classificador, as curvas para NO_HIS (Base sem histórico), BIN_ALL (todos os atributos com representação binária), TF_ALL (todos os atributos com representação TF) e TF_IDF_ALL (todos os atributos com representação TF-IDF). A curva precision-recall é um bom indicador da qualidade dos classificadores quando a base tem um alto fator de desbalanceamento, como é o caso da base analisada neste trabalho. Na plotagem é possível observar que os resultados para a classe S estão todas próximas ao ideal (a curva se aproxima ao canto superior direito do gráfico, com área sob a curva próxima a 1). Já para a classe N há mais variações, com os piores resultados pertencentes ao dados sem histórico (NO_HIS).

Na Figura 18, a melhor avaliação do PRC é para o resultado MelhorRF (melhor

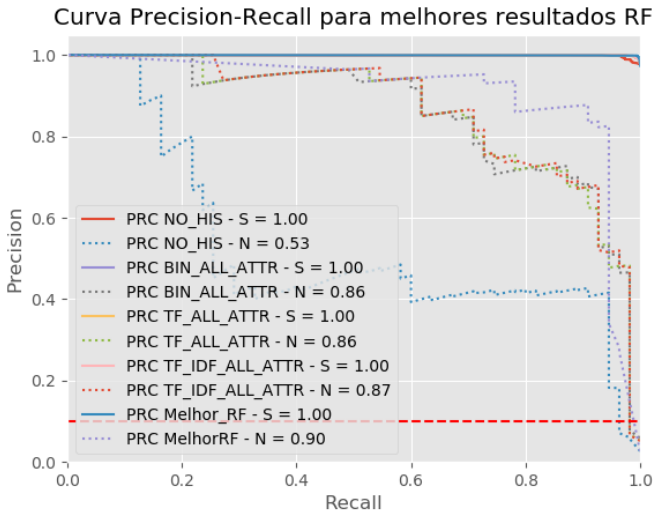

Figura 18 – PRC para o melhor resultado RF

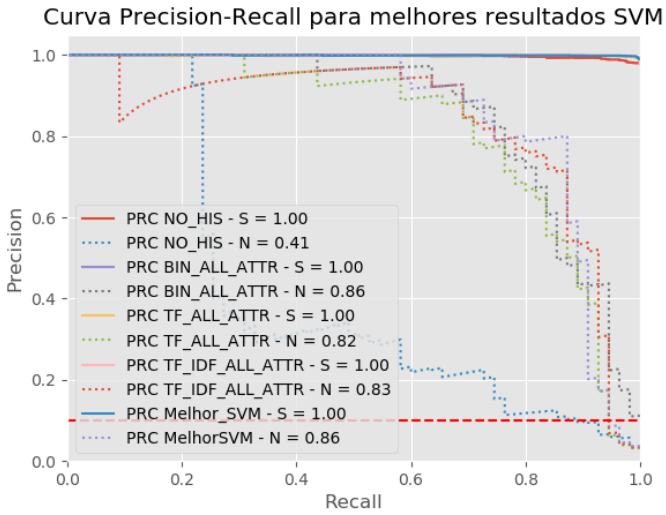

Figura 19 – PRC para o melhor resultado SVM

resultado do classificador RF para uma técnica de seleção de atributos). A pior é a relacionada aos dados sem histórico (NO_HIS). O mesmo se repete para os outros gráficos, como por exemplo, na Figura 19, em que os resultados da MelhorSVM são idênticos aos do histórico com todos os atributos e representação binária (BIN_ALL). Na Figura 20, referente aos resultados do KNN, é possível observar que os resultados NO_HIS para esse classificador é melhor que nos outros com valor de PRC para Classe N de 0.63. Também é possível observar que os resultados para o classificador KNN apresentaram menos variação em relação aos outros classificadores. As curvas PRC da Classe N para o classificador NB são as que possuem maior variação. Como foi dito antes, o classificador NB apresentou resultados pouco significativos em todos os testes feitos.

Por fim, foi realizado o teste de hipótese dos melhores resultados para cada clas-

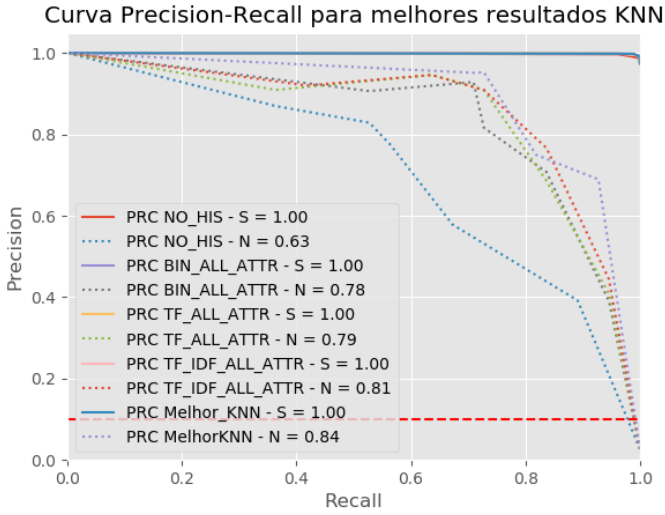

Figura 20 – PRC para o melhor resultado KNN

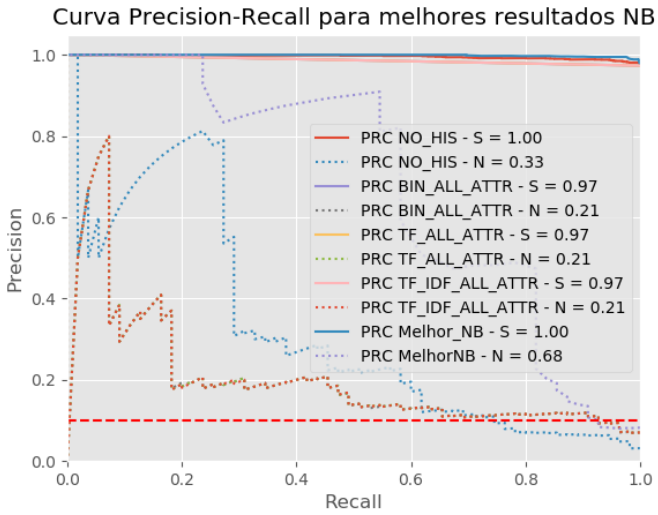

Figura 21 – PRC para o melhor resultado NB

Tabela 49 – Teste de hipótese melhores resultados dos classificadores para seleção de atributos

Teste de Hipótese	**Z**	$Z_{\frac{\alpha}{2}}$	**Z**>$Z_{\frac{\alpha}{2}}$	**Rejeita** H_0
NO_HIS vs Melhor RF	4.93515674616159	1.96	Verdadeiro	Verdadeiro
NO_HIS vs Melhor SVM	3.974424599247822	1.96	Verdadeiro	Verdadeiro
NO_HIS vs Melhor KNN	2.04056946215298	1.96	Verdadeiro	Verdadeiro
NO_HIS vs Melhor NB	4.1975013032982655	1.96	Verdadeiro	Verdadeiro

sificador. Para este teste foi considerado um nível de significância de 5% ($Z_{\frac{\alpha}{2}} = \pm 1.96$). Como foi definido, as hipóteses testadas pela pesquisa são:

- Hipótese nula: não há diferença no aprendizado de máquina durante o processo de regulação, quando se usa informações do histórico de atendimentos de um beneficiário.
- Hipótese alternativa: existe diferença no aprendizado de máquina durante o pro-

cesso de regulação, quando se usa informações do histórico de atendimentos de um beneficiário.

No teste de hipótese Z (Equação 3.38) é verificada se a diferença de resultados entre dois classificadores são estatisticamente diferentes considerando um nível de significância. No teste de hipótese proposto, é verificada a hipótese nula, onde é calculado se os resultados de índices Kappas dos resultados sem uso de histórico (NO_HIS) e os demais resultados das técnicas aplicadas. Dessa forma se o valor Z calculado não ultrapassar os limites de $\pm$ 1,96, a hipótese nula é aceita, caso contrário ela é rejeitada. Os resultados para o teste hipótese podem ser vistos na Tabela 49, nessa tabela são testados valores estatisticamente superiores ($Z > Z_{\frac{\alpha}{2}}$), pois um dos objetivos da pesquisa é demonstrar que a metodologia proposta consegue melhorar os resultados da aprendizagem.

Conforme demonstrado na Tabela 49, os resultados do teste de hipótese apontam que há diferença significativa entre as classificações, uma vez que o uso do histórico apresenta resultados estatisticamente superiores em relação ao não uso do histórico.

5.2 Transformação de Atributos

Nesta etapa dos testes foram aplicados quatro algoritmos clássicos de transformação de dados: PCA, KPCA, ICA e LSA. Para cada uma dessas técnicas são testados os algoritmos de classificação RF, SVM, KNN e NB com as representação Binária (BIN), TF e TF-IDF. Os resultados são mostrados adiante.

5.2.1 Principal Component Analysis - PCA

A implementação de PCA utilizada também foi a presente no WEKA. As componentes escolhidas foram aquelas que no mínimo acumulassem 95% de variância. A Tabela 50 aponta que os resultados obtidos com o PCA foram superiores quando comparados ao resultado NO_HIS. Porém os resultados obtidos com PCA não conseguiram superar os resultados com todos os atributos (BIN_ALL, TF_ALL, TF_IDF_ALL). Na Tabela 51, têm-se resultados parecidos aos obtidos com a classificação RF, porém um pouco melhores. Nos dois casos a representação Binária com uso do PCA obteve os melhores resultados comparado às outras representações combinadas com PCA. Porém, os melhores resultados foram aqueles que utilizaram todos os atributos do histórico (ALL).

Para o classificador KNN (Tabela 52), nenhuma das combinações BIN, TF e TF_IDF com PCA conseguiram superar os resultados sem histórico (NO_HIS). Ao olhar os resultados da Tabela 53, pode-se perceber que o classificador NB continua a apresentar resultados inexpressivos, porém, ainda assim, melhores que os resultados NO_HIS, BIN_ALL, TF_ALL, TF_IDF_ALL.

Tabela 50 – Resultado do classificador RF para a técnica de transformação de atributos PCA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.97	0.3255	0.98	0.41	0.99	0.29
BIN_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_IDF_ALL	4553	0.9894	0.7746	0.99	0.87	1.0	0.71
BIN_PCA	589	0.9792	0.4468	0.98	0.75	1.0	0.33
TF_PCA	585	0.9782	0.4066	0.98	0.73	1.0	0.29
TF_IDF_PCA	585	0.9782	0.4066	0.98	0.73	1.0	0.29

Tabela 51 – Resultado do classificador SVM para a técnica de transformação de atributos PCA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9744	0.3181	0.98	0.54	0.99	0.24
BIN_ALL	4553	0.9898	0.7869	0.99	0.87	1.0	0.73
TF_ALL	4553	0.9894	0.7701	0.99	0.88	1.0	0.69
TF_IDF_ALL	4553	0.9884	0.7588	0.99	0.83	1.0	0.71
BIN_PCA	589	0.9797	0.5431	0.99	0.67	0.99	0.47
TF_PCA	585	0.9777	0.3857	0.98	0.71	1.0	0.27
TF_IDF_PCA	585	0.9777	0.429	0.98	0.67	1.0	0.33

Tabela 52 – Resultado do classificador KNN para a técnica de transformação de atributos PCA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.984	0.6447	0.99	0.78	1.0	0.56
BIN_ALL	4553	0.9884	0.7633	0.99	0.82	1.0	0.73
TF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
TF_IDF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
BIN_PCA	589	0.9797	0.513	0.98	0.7	1.0	0.42
TF_PCA	585	0.9768	0.4042	0.98	0.63	1.0	0.31
TF_IDF_PCA	585	0.9753	0.4021	0.98	0.56	0.99	0.33

Tabela 53 – Resultado do classificador NB para a técnica de transformação de atributos PCA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_PCA	589	0.894	0.1915	0.99	0.14	0.9	0.58
TF_PCA	585	0.9178	0.2362	0.99	0.18	0.93	0.56
TF_IDF_PCA	585	0.9211	0.2385	0.99	0.18	0.93	0.55

5.2.2 Kernel PCA - KPCA

O Kernel PCA utilizado foi o implementado pelo WEKA, onde as componentes escolhidas foram aquelas cuja a variância acumularam no mínimo 95%. Os resultados apresentados pela Tabela 54 correspondem a aplicação do algoritmo RF sobre os dados transformados pelo KPCA. Comparando esses resultados com os obtidos pelo RF com transformação PCA, têm-se resultados melhores com KPCA. Porém ao olhar para o

resultado da Tabela 55, e os obtidos pelo SVM com PCA (Tabela 51), têm-se melhores resultados com PCA. Ou seja, para o classificador SVM a transformação PCA foi melhor do que a KPCA para representar os dados.

Tabela 54 – Resultado do classificador RF para a técnica de transformação de atributos KPCA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.97	0.3255	0.98	0.41	0.99	0.29
BIN_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_IDF_ALL	4553	0.9894	0.7746	0.99	0.87	1.0	0.71
BIN_KPCA	206	0.9874	0.7229	0.99	0.85	1.0	0.64
TF_KPCA	206	0.984	0.6296	0.99	0.81	1.0	0.53
TF_IDF_KPCA	206	0.9845	0.6446	0.99	0.81	1.0	0.55

Tabela 55 – Resultado do classificador SVM para a técnica de transformação de atributos KPCA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9744	0.3181	0.98	0.54	0.99	0.24
BIN_ALL	4553	0.9898	0.7869	0.99	0.87	1.0	0.73
TF_ALL	4553	0.9894	0.7701	0.99	0.88	1.0	0.69
TF_IDF_ALL	4553	0.9884	0.7588	0.99	0.83	1.0	0.71
BIN_KPCA	206	0.9734	0.0	0.97	0.0	1.0	0.0
TF_KPCA	206	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_KPCA	206	0.9734	0.0	0.97	0.0	1.0	0.0

Para a transformação KPCA, os resultados obtidos com KNN(Tabela 56) são os mais promissores comparados aos outros algoritmos de classificação testados. Nesse teste os melhores resultados foram aqueles que utilizaram a representação TF_IDF. Para o classificador NB (Tabela 57) o melhor resultado também foi aquele que utilizou a representação TF_IDF combinada com o KPCA.

Tabela 56 – Resultado do classificador KNN para a técnica de transformação de atributos KPCA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.984	0.6447	0.99	0.78	1.0	0.56
BIN_ALL	4553	0.9884	0.7633	0.99	0.82	1.0	0.73
TF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
TF_IDF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
BIN_KPCA	206	0.9884	0.7633	0.99	0.82	1.0	0.73
TF_KPCA	206	0.9884	0.7588	0.99	0.83	1.0	0.71
TF_IDF_KPCA	206	0.9889	0.7666	0.99	0.85	1.0	0.71

Tabela 57 – Resultado do classificador NB para a técnica de transformação de atributos KPCA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_KPCA	206	0.7702	0.1106	0.99	0.08	0.77	0.78
TF_KPCA	206	0.8379	0.1414	0.99	0.1	0.84	0.67
TF_IDF_KPCA	206	0.924	0.3093	0.99	0.22	0.93	0.73

5.2.3 Independent Component Analysis - ICA

O algoritmo ICA testado também pertence ao WEKA. Ele foi configurado para rodar 2000 iterações caso o erro não convergisse para 0.001. A Tabela 58 apresenta o resultado do classificador RF para a transformação ICA. Os resultados usando a representação TF e TF_IDF foram melhores que a Binária. Os resultados da Tabela 59 e 61 apresentam o resultado da classificação SVM e NB respectivamente. A partir da observação das tabelas, percebe-se que a transformação ICA não foi bem sucedida em classificar a base com histórico. Uma exceção é o resultado da combinação BIN_ICA para o classificador NB, que foi o único resultado que conseguiu classificar a classe N.

O classificador KNN (Tabela 60 apresentou os melhores para a transformação ICA. Onde o melhor resultado foi utilizando a representação TF_IDF.

Tabela 58 – Resultado do classificador RF para a técnica de transformação de atributos ICA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.97	0.3255	0.98	0.41	0.99	0.29
BIN_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_IDF_ALL	4553	0.9894	0.7746	0.99	0.87	1.0	0.71
BIN_ICA	85	0.9744	0.2013	0.98	0.58	1.0	0.13
TF_ICA	10	0.9826	0.5915	0.99	0.77	1.0	0.49
TF_IDF_ICA	10	0.9845	0.6446	0.99	0.81	1.0	0.55

Tabela 59 – Resultado do classificador SVM para a técnica de transformação de atributos ICA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9744	0.3181	0.98	0.54	0.99	0.24
BIN_ALL	4553	0.9898	0.7869	0.99	0.87	1.0	0.73
TF_ALL	4553	0.9894	0.7701	0.99	0.88	1.0	0.69
TF_IDF_ALL	4553	0.9884	0.7588	0.99	0.83	1.0	0.71
BIN_ICA	85	0.9734	0.0	0.97	0.0	1.0	0.0
TF_ICA	10	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_ICA	10	0.9734	0.0	0.97	0.0	1.0	0.0

Tabela 60 – Resultado do classificador KNN para a técnica de transformação de atributos ICA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.984	0.6447	0.99	0.78	1.0	0.56
BIN_ALL	4553	0.9884	0.7633	0.99	0.82	1.0	0.73
TF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
TF_IDF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
BIN_ICA	85	0.9821	0.6316	0.99	0.69	0.99	0.6
TF_ICA	10	0.985	0.6283	0.99	0.9	1.0	0.49
TF_IDF_ICA	10	0.9894	0.7504	0.99	0.97	1.0	0.62

Tabela 61 – Resultado do classificador NB para a técnica de transformação de atributos ICA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_ICA	85	0.9782	0.5906	0.99	0.59	0.99	0.62
TF_ICA	10	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_ICA	10	0.9734	0.0	0.97	0.0	1.0	0.0

5.2.4 Latent Sematic Analysis - LSA

O algoritmo LSA foi testado utilizando a ferramenta WEKA. Ao observar os resultados das Tabelas 62, 63, 64, 65, percebe-se que os classificadores SVM e NB para essa transformação não foram bem sucedidos em classificar a classe minoritária N. Para a classificação RF, a melhor combinação foi a representação binária com LSA (BIN_LSA). Já para a classificação KNN a melhor combinação usando a transformação LSA foi a TF_IDF_LSA.

Tabela 62 – Resultado do classificador RF para a técnica de transformação de atributos LSA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.97	0.3255	0.98	0.41	0.99	0.29
BIN_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_ALL	4553	0.9889	0.7666	0.99	0.85	1.0	0.71
TF_IDF_ALL	4553	0.9894	0.7746	0.99	0.87	1.0	0.71
BIN_LSA	2	0.9884	0.7835	1.0	0.76	0.99	0.82
TF_LSA	2	0.9894	0.7789	0.99	0.85	1.0	0.73
TF_IDF_LSA	2	0.9884	0.7759	0.99	0.78	0.99	0.78

Tabela 63 – Resultado do classificador SVM para a técnica de transformação de atributos LSA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9744	0.3181	0.98	0.54	0.99	0.24
BIN_ALL	4553	0.9898	0.7869	0.99	0.87	1.0	0.73
TF_ALL	4553	0.9894	0.7701	0.99	0.88	1.0	0.69
TF_IDF_ALL	4553	0.9884	0.7588	0.99	0.83	1.0	0.71
BIN_LSA	2	0.9734	0.0	0.97	0.0	1.0	0.0
TF_LSA	2	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_LSA	2	0.9734	0.0	0.97	0.0	1.0	0.0

Tabela 64 – Resultado do classificador KNN para a técnica de transformação de atributos LSA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.984	0.6447	0.99	0.78	1.0	0.56
BIN_ALL	4553	0.9884	0.7633	0.99	0.82	1.0	0.73
TF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
TF_IDF_ALL	4553	0.9908	0.8034	0.99	0.91	1.0	0.73
BIN_LSA	2	0.985	0.713	0.99	0.71	0.99	0.73
TF_LSA	2	0.9874	0.7336	0.99	0.82	1.0	0.67
TF_IDF_LSA	2	0.9874	0.7436	0.99	0.8	1.0	0.71

Tabela 65 – Resultado do classificador NB para a técnica de transformação de atributos LSA

Técnica	N. Atributos	Acurácia	Kappa	Precisão		Recall	
				Classe S	Classe N	Classe S	Classe N
NO_HIS	393	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_ALL	4553	0.9734	0.0	0.97	0.0	1.0	0.0
BIN_LSA	2	0.9734	0.0	0.97	0.0	1.0	0.0
TF_LSA	2	0.9734	0.0	0.97	0.0	1.0	0.0
TF_IDF_LSA	2	0.9734	0.0	0.97	0.0	1.0	0.0

5.2.5 Melhores resultados transformação de atributos

Apesarem de serem efetivos em diminuir a dimensionalidade dos dados, as técnicas de transformação de atributos tiveram resultados inferiores quando comparados com a seleção de atributos. Os melhores resultados presentes na Tabela 66 demonstram que a técnica LSA obteve os melhores resultados quando aplicado ao RF, em comparação às demais técnicas. O KNN vem em seguida com representação TF_IDF_KPCA. Ao analisar o resultado do teste de hipótese na Tabela 67, têm-se que somente os resultados RF_BIN_LSA e o NB_BIN_ICA passam pelo teste, enquanto os resultados obtidos pelo KNN e SVM se aproximam da margem de 1,96 mas não conseguem ultrapassá-la. Mesmo o KNN apresentando a segunda melhor estimativa dos testes com transformação de atributos, a diferença entre os resultados apresentados por ele em relação ao NO_HIS é pouca. Esse resultado demonstra que os classificadores KNN e SVM, para o a modelagem com transformação de atributos, não puderam apresentar um modelo de classificação

satisfatório. Outra observação a ser feita é que a representação binária (BIN) para os algoritmos de transformação de atributos se saíram melhores que as demais representações.

Tabela 66 – Tabela com os melhores resultados de para cada classificador usando transformação de atributos

Classificador	**Representação dos dados**	**N. Atributos**	**Acurária**	**Kappa**	**Precisão S**	**Precisão N**	**Recall S**	**Recall N**
RF	BIN_LSA	2	0.9894	0.7789	0.99	0.85	1.0	0.73
SVM	BIN_PCA	589	0.9797	0.5431	0.99	0.67	0.99	0.47
KNN	TF_IDF_KPCA	206	0.9889	0.7666	0.99	0.85	1.0	0.71
NB	BIN_ICA	85	0.9782	0.5906	0.99	0.59	0.99	0.62

Tabela 67 – Teste de hipótese melhores resultados dos classificadores para transformação de atributos

Teste de Hipótese	**Z**	$Z_{\frac{\alpha}{2}}$	**Z>**$Z_{\frac{\alpha}{2}}$	**Rejeita** H_0
NO_HIS vs RF_BIN_LSA	4.161286744648549	1.96	Verdadeiro	Verdadeiro
NO_HIS vs Melhor SVM_BIN_PCA	1.8551904933480008	1.96	Falso	Falso
NO_HIS vs Melhor KNN_TF_IDF_KPCA	1.4425841823444012	1.96	Falso	Falso
NO_HIS vs Melhor NB_BIN_ICA	3.9688933478272	1.96	Verdadeiro	Verdadeiro

As curvas PRC presentes nas Figuras 22, 23, 24 e 25 apresentam o resultado dos classificadores RF, SVM, KNN e NB respectivamente. Para cada gráfico são apresentados o resultado das curvas PRC para a Classe S e N. Assim como nos resultados por seleção de atributos, as curvas referentes à classe S estão próximas ao ideal. Porém os resultados para a classe N variam muito. Para a Figura 22, com os resultados dos classificador RF, é possível observar que a curva referente ao resultado PRC_BIN_LSA está bem próxima do melhor resultado (TF_IDF_ALL). A pior estimativa da curva PRC é a NO_HIS (sem histórico). A curva presente na Figura 23 que representa a estimativa da classe N para a combinação BIN_PCA não perde somente para a representação NO_HIS. Para a Figura 24, pode-se observar que as curvas de cada uma das combinações estão muito próximas umas das outras. Isso significa que os resultados dessas combinações pouco diferiram entre si, isso explica, por exemplo, o fato do teste TF_IDF_KPCA não ter passado no teste de hipótese. Por fim, para a Figura 25, têm-se os valores das estimativas do classificador NB. Pelo gráfico é possível observar que o classificador apresentou dificuldades para predizer o atributo N. Das combinações apresentadas no gráfico, somente o BIN_ICA apresentou o melhor resultado.

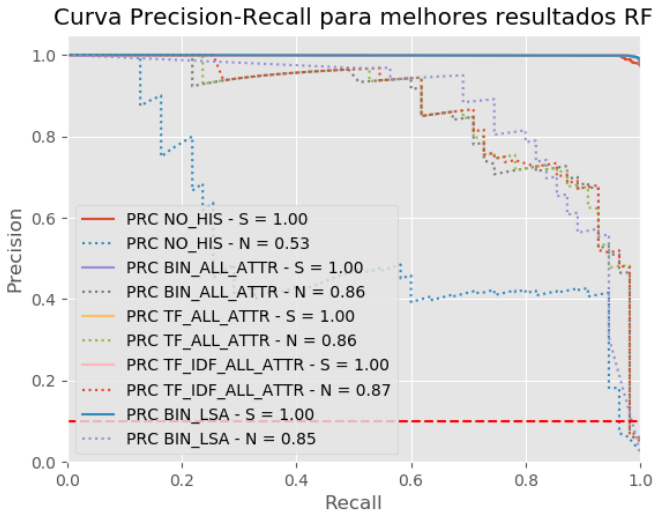

Figura 22 – PRC para o melhor resultado RF

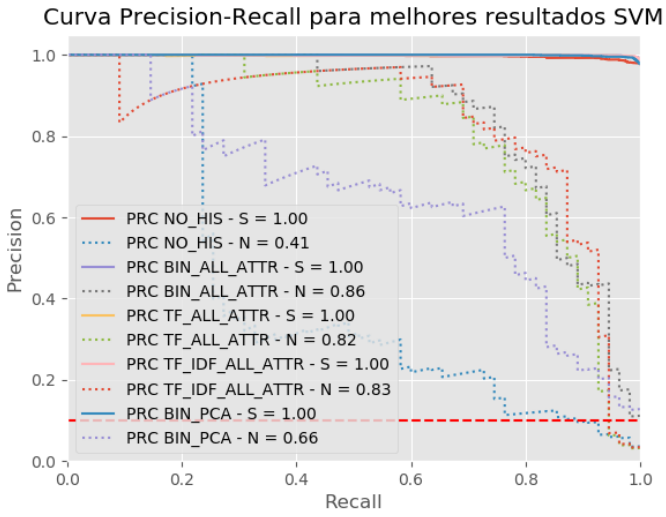

Figura 23 – PRC para o melhor resultado SVM

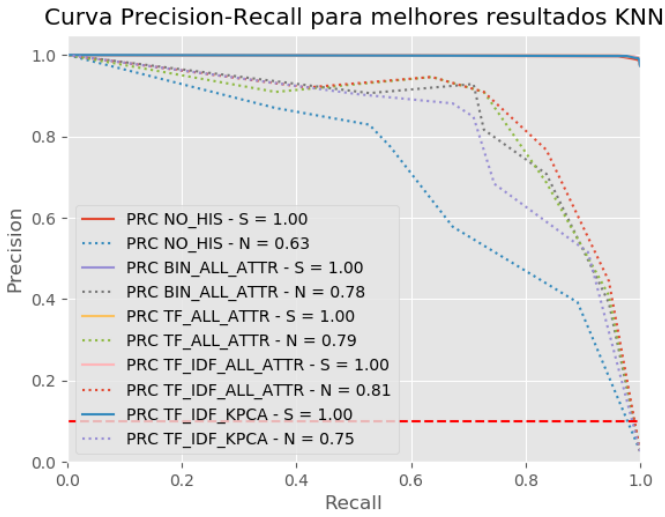

Figura 24 – PRC para o melhor resultado KNN

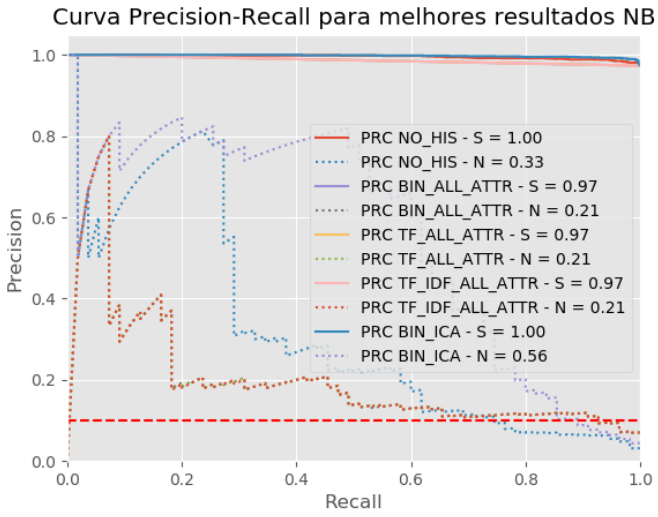

Figura 25 – PRC para o melhor resultado NB

5.3 Considerações Finais

Este capítulo apresentou uma descrição dos experimentos e resultados obtidos com a aplicação do histórico ao processo de regulação em uma OPS. Foram realizados testes comparativos entre diferentes técnicas de seleção, transformação de atributos combinadas com as representações de dados binário, TF e TF-IDF, de modo que o resultado obtido pudesse demonstrar a viabilidade do uso de histórico como informação auxiliar para o problema em questão. A partir do teste de hipótese Z e análise das métricas de avaliação, é possível afirmar que o uso do histórico trouxe uma melhora significativa à aprendizagem, quando comparado com a aprendizagem da regulação sem histórico.

No próximo capítulo serão apresentados a conclusão e trabalhos futuros deste estudo.

6 Conclusões e Trabalhos Futuros

6.1 Conclusões

Este trabalho apresenta um estudo do uso de informações do histórico de beneficiários de maneira a melhorar a aprendizagem em regulação de um OPS. Para isso, foi apresentada uma proposta baseada na modelagem e preparação do histórico com diferentes aplicações de técnicas de seleção e transformação de atributos. Cada combinação de representação de dados (Binária, TF e TF-IDF) com técnicas de seleção e transformação de atributos foram testadas em quatro classificadores: RF, SVM, KNN e NB. Após a aplicação dos classificadores são avaliados os resultados obtidos, comparando com os resultados sem o uso de histórico. A partir das informações apresentados no Capítulo 5, é possível responder à questão levantada pela pesquisa:

- **Q1:** A adição da informação de histórico de beneficiários auxilia o processo de regulação em uma OPS?
  R: Os resultados apontam que a modelagem da base com histórico apresentou melhora na classificação para todos os classificadores testados. Onde, para as técnicas de seleção de atributos, os algoritmos IG (*Information Gain*) e GR (*Gain Ratio*) combinados com as representações TF e TF-IDF apresentaram os melhores resultados no classificador RF. Para a transformação de atributos o melhor resultado foi utilização representação Binária com LSA para o algoritmo RF também. É importante destacar que, a grande maioria dos resultados obtidos, obtiveram indicadores superiores em acurácia, índice Kappa, Precisão e *Recall*, com resultados estatisticamente significativos quando comparados aos obtidos com a base sem histórico.

Com relação aos objetivos geral e específico definidos pela pesquisa, e, a partir dos resultados obtidos temos que:

- **Objetivo Geral:** melhorar o aprendizado automático do processo de regulação médica, por meio do uso do histórico de beneficiários.
  R: A partir dos resultados obtidos e avaliação do teste de hipótese Z, é possível concluir que houve, sim, melhora no aprendizado com uso do histórico para a base utilizada.

- **Objetivos Específicos:**
  Revisão da literatura e aplicação de diferentes técnicas de seleção, transformação e representação dos dados.

R: A partir da análise de trabalhos relacionados ao tema foram selecionados alguns dos principais algoritmos de seleção de atributos: CFS, CSE, WSE, IG, GR e Relief; e transformação de atributos: PCA, KPCA, ICA e LSA; e, para representação das informações: Binária, TF e TF-IDF.

Aplicação de algoritmos de classificação evidenciando quais modelagens influenciaram positivamente nos resultados:

R: Foram utilizados os algoritmos RF, SVM, KNN e NB para cada uma das técnicas utilizadas comparando os resultados com a base sem o uso do histórico. A fim de analisar se os resultados obtidos foram estatisticamente significativos, foi realizado o teste de hipótese Z.

Realização de customização dos modelos de maneira a melhorar os resultados:

R: A fim de melhorar os resultados foram analisadas diversas combinações de técnicas de seleção e transformação de atributos com as diferentes técnicas de representação de dados.

Baseado nessas informações é possível apontar as seguinte contribuições da pesquisa:

- Apresenta uma proposta de modelagem do histórico de solicitações de beneficiários de uma OPS;
- Apresenta alternativas para otimização dos resultados por meio do uso de seleção e transformação de atributos;
- Apresenta o potencial que o uso do histórico traz à aprendizagem automática da regulação médica.

As principais limitações da abordagem proposta são: a base utilizada, que apresenta poucos registros e um alto fator de desbalanceamento entre as classes; a utilização de outras bases para realizar uma comparação mais ampla que fortalecesse ainda mais a conclusão de que o histórico traz benefícios para aprendizagem; a dificuldade em se encontrar trabalhos relacionados a modelagem do histórico para efeito comparativo com a bordagem proposta.

Um artigo detalhando este trabalho foi publicado na 8th *Brazilian Conference on Intelligent Systems* (Bracis 2019), sob o título *Using Historical Information of Patients for Prior Authorization Learning.*

6.2 Trabalhos Futuros

Apesar dos resultados serem prósperos, eles ainda podem ser melhorados, sendo necessário o estudo e aplicação de mais técnicas para obtenção de melhores resultados. Como trabalhos futuros têm-se:

- A realização de mais testes com novas bases de dados;
- Ampliar os estudos para melhorar o processo de modelagem do histórico de beneficiários na etapa de preparação da base;
- Aplicação de técnicas de transformação de atributos baseadas em *Deep Learning* como *Convolutional Neural Network*. Diversos trabalhos na literatura provaram a capacidade desse tipo de rede em se extrair informações dos mais variados tipos de dados com ganhos significativos para a aprendizagem.

Referências

ALAM, T. M. et al. A model for early prediction of diabetes. *Informatics in Medicine Unlocked*, v. 16, p. 100204, 2019. ISSN 2352-9148. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2352914819300176>. Citado na página 4.

AMB. *Classificação Brasileira Hierarquizada de Procedimentos Médicos*. 2016. <https://amb.org.br/cbhpm/>. [Online; Accessed em 31/04/2018]. Citado na página 52.

ANS. *Agência Nacional de Saúde Suplementar*. 2018. <http://www.ans.gov.br/>. [Online; Accessed em 21/06/2018]. Citado na página 1.

ANS. *Caderno de Informação da Saúde Suplementar*. 2018. <http://www.ans.gov.br/images/stories/Materiais_para_pesquisa/Perfil_setor/Caderno_informacao_saude_suplementar/caderno_informacao_junho_2017.pdf>. [Online; Accessed em 21/06/2018]. Citado 3 vezes nas páginas 9, 1 e 2.

ANTONELLI, D. et al. Analysis of diabetic patients through their examination history. *Expert Systems with Applications*, Elsevier, v. 40, n. 11, p. 4672–4678, 2013. Citado na página 13.

ARAUJO, F. *Descoberta de conhecimento em base de dados para o aprendizado da regulação médica/odontológica em operadora de plano de saúde*. 2014. Citado 4 vezes nas páginas 2, 42, 44 e 45.

ARAÚJO, F. H.; SANTANA, A. M.; NETO, P. d. A. S. Using machine learning to support healthcare professionals in making preauthorisation decisions. *International journal of medical informatics*, Elsevier, v. 94, p. 1–7, 2016. Citado na página 11.

BARROS, J. L. de; BEIRUTH, A. X. Aplicação de modelos de previsão de insolvência nas operadoras de planos de saúde do brasil. *RAGC*, v. 4, n. 15, 2016. Citado na página 2.

BRASIL. *LEI Nº 9.656, DE 3 DE JUNHO DE 1998. Dispõe sobre os planos e seguros privados de assistência à saúde*. 1998. <http://www.planalto.gov.br/ccivil_03/Leis/L9656.htm>. [Online; Accessed em 21/06/2018]. Citado na página 1.

BRASIL. *LEI No 9.961 DE 28 DE JANEIRO DE 2000. Cria a Agência Nacional de Saúde Suplementar – ANS e dá outras providências.* 2000. <http://www.planalto.gov.br/ccivil_03/Leis/L9961.htm/>. [Online; Accessed em 21/06/2018]. Citado na página 1.

BRASIL. *Sistema Único de Saúde*. 2018. <http://portalms.saude.gov.br/sistema-unico-de-saude>. [Online; Accessed em 21/06/2018]. Citado na página 1.

CARVALHO, R. R. P.; FORTES, P. A. de C.; GARRAFA, V. A saúde suplementar em perspectiva bioética. *Revista da Associação Médica Brasileira*, Elsevier, v. 59, n. 6, p. 600–606, 2013. Citado na página 1.

CHIMIESKI, B. F.; FAGUNDES, R. D. R. Association and classification data mining algorithms comparison over medical datasets. *Journal of health informatics*, v. 5, n. 2, 2013. Citado 2 vezes nas páginas 13 e 45.

COMON, P. Independent component analysis, a new concept? *Signal Processing*, v. 36, n. 3, p. 287 – 314, 1994. ISSN 0165-1684. Higher Order Statistics. Disponível em: <http://www.sciencedirect.com/science/article/pii/0165168494900299>. Citado na página 36.

CONGALTON, R. G.; GREEN, K. *Assessing the accuracy of remotely sensed data: principles and practices*. [S.l.]: CRC press, 2008. Citado na página 47.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine learning*, Springer, v. 20, n. 3, p. 273–297, 1995. Citado na página 44.

COVER, T. M. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, IEEE, n. 3, p. 326–334, 1965. Citado na página 44.

ESTER, M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Kdd*. [S.l.: s.n.], 1996. v. 96, n. 34, p. 226–231. Citado na página 13.

FAN, S.-K. S. et al. Using machine learning and big data approaches to predict travel time based on historical and real-time data from taiwan electronic toll collection. *Soft Computing*, v. 22, n. 17, p. 5707–5718, Sep 2018. ISSN 1433-7479. Disponível em: <https://doi.org/10.1007/s00500-017-2610-y>. Citado na página 4.

FIX, E.; JR, J. L. H. *Discriminatory analysis-nonparametric discrimination: consistency properties*. [S.l.], 1951. Citado na página 45.

GOLDBERG, D. E.; HOLLAND, J. H. Genetic algorithms and machine learning. *Machine learning*, Springer, v. 3, n. 2, p. 95–99, 1988. Citado na página 20.

GOMES, A. K. *Análise do conhecimento extraído de classificadores simbólicos utilizando medidas de avaliação e de interessabilidade.* 2002. Citado na página 46.

GUPTA, V.; LEHAL, G. S. et al. A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence*, v. 1, n. 1, p. 60–76, 2009. Citado na página 15.

GUTLEIN, M. et al. Large-scale attribute selection using wrappers. In: IEEE. *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*. [S.l.], 2009. p. 332–339. Citado na página 20.

HALL, M. A. Correlation-based feature selection for machine learning. University of Waikato Hamilton, 1999. Citado na página 23.

HALL, M. A.; HOLMES, G. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data engineering*, IEEE, v. 15, n. 6, p. 1437–1447, 2003. Citado 2 vezes nas páginas 20 e 26.

HAYKIN, S. *Neural networks: a comprehensive foundation*. [S.l.]: Prentice Hall PTR, 1994. Citado na página 44.

HU, Q.; ZHANG, R.; ZHOU, Y. Transfer learning for short-term wind speed prediction with deep neural networks. *Renewable Energy*, v. 85, p. 83 – 95, 2016. ISSN 0960-1481. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0960148115300574>. Citado na página 4.

Hyvarinen, A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, v. 10, n. 3, p. 626–634, May 1999. ISSN 1045-9227. Citado na página 38.

INFOWAY. *INFOWAY e-health company*. 2018. Disponível em: <https://www.infoway-br.com/>. Citado na página 51.

JOHN, G. H.; LANGLEY, P. Estimating continuous distributions in bayesian classifiers. In: MORGAN KAUFMANN PUBLISHERS INC. *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. [S.l.], 1995. p. 338–345. Citado na página 43.

KHAMIS, H. S.; CHERUIYOT, K. W.; KIMANI, S. Application of k-nearest neighbour classification in medical data mining. *International Journal of Information and Communication Technology Research*, v. 4, n. 4, 2014. Citado na página 4.

KHANDANI, A. E.; KIM, A. J.; LO, A. W. Consumer credit-risk models via machine-learning algorithms. *Journal of Banking and Finance*, v. 34, n. 11, p. 2767 – 2787, 2010. ISSN 0378-4266. Citado na página 4.

KHARYA, S. Using data mining techniques for diagnosis and prognosis of cancer disease. *arXiv preprint arXiv:1205.1923*, 2012. Citado na página 13.

KIANPISHEH SAEED JALILI, N. M. C. S. Predicting Job Wait Time in Grid Environment by Applying Machine Learning Methods on Historical Information. *International Journal of Grid and Distributed Computing*, 5, p. 11–22, 2012. Disponível em: <https://www.earticle.net/Article/A208073>. Citado na página 4.

KOHAVI, R.; JOHN, G. H. Wrappers for feature subset selection. *Artificial intelligence*, Elsevier, v. 97, n. 1-2, p. 273–324, 1997. Citado 2 vezes nas páginas 19 e 26.

KONONENKO, I. Estimating attributes: analysis and extensions of relief. In: SPRINGER. *European conference on machine learning*. [S.l.], 1994. p. 171–182. Citado na página 25.

KUMAR, V.; MINZ, S. Feature selection. *SmartCR*, v. 4, n. 3, p. 211–229, 2014. Citado 3 vezes nas páginas 9, 16 e 17.

LANDIS, J. R.; KOCH, G. G. The measurement of observer agreement for categorical data. *biometrics*, JSTOR, p. 159–174, 1977. Citado na página 47.

LIU, H.; SETIONO, R. et al. A probabilistic approach to feature selection-a filter solution. In: CITESEER. *ICML*. [S.l.], 1996. v. 96, p. 319–327. Citado na página 23.

LUCINI, F. R. et al. Text mining approach to predict hospital admissions using early medical records from the emergency department. *International Journal of Medical Informatics*, v. 100, p. 1 – 8, 2017. ISSN 1386-5056. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1386505617300011>. Citado na página 12.

MARINS, O. L. F. et al. Aplicação de algoritmos de aprendizagem de máquina para mineração de dados sobre beneficiários de planos de saúde suplementar. *Journal of Health Informatics*, v. 4, n. 2, 2012. Citado na página 10.

MEDEIROS, M. L. As falhas de mercado e os mecanismos de regulação da saúde suplementar no brasil com uma abordagem das novas responsabilidades para os agentes desse mercado. 2012. Citado na página 1.

MENAHEM, E.; PUSIZ, R.; ELOVICI, Y. Detecting spammers via aggregated historical data set. In: XU, L.; BERTINO, E.; MU, Y. (Ed.). *Network and System Security*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 248–262. ISBN 978-3-642-34601-9. Citado na página 4.

MIOTTO, R. et al. Deep patient: An unsupervised representation to predict the future of patients from the electronic health records. *Scientific Reports*, v. 6, n. 1, p. 26094, 2016. Disponível em: <https://app.dimensions.ai/details/publication/pub.1013280079andhttps://www.nature.com/articles/srep26094.pdf>. Citado na página 13.

MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997. ISBN 0070428077, 9780070428072. Citado na página 42.

MOTODA, H.; LIU, H. Feature selection, extraction and construction. *Communication of IICM (Institute of Information and Computing Machinery, Taiwan) Vol*, v. 5, p. 67–72, 2002. Citado 2 vezes nas páginas 17 e 27.

PYTHON. *Python Software Foundation version 3.7*. 2019. Disponível em: <https://www.python.org/>. Citado na página 51.

QUINLAN, J. R. Induction of decision trees. *Machine learning*, Springer, v. 1, n. 1, p. 81–106, 1986. Citado 3 vezes nas páginas 10, 20 e 22.

Richards, L. et al. Comparing classifiers in historical census linkage. In: *2014 IEEE International Conference on Data Mining Workshop*. [S.l.: s.n.], 2014. p. 1086–1094. ISSN 2375-9232. Citado na página 4.

ROSENFIELD, G. H.; FITZPATRICK-LINS, K. A coefficient of agreement as a measure of thematic classification accuracy. *Photogrammetric engineering and remote sensing*, v. 52, n. 2, p. 223–227, 1986. Citado na página 47.

SALTON, G.; BUCKLEY, C. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 24, n. 5, p. 513–523, ago. 1988. ISSN 0306-4573. Disponível em: <http://dx.doi.org/10.1016/0306-4573(88)90021-0>. Citado 2 vezes nas páginas 15 e 16.

SALTON, G.; MCGILL, M. J. *Introduction to modern information retrieval*. [S.l.]: mcgraw-hill, 1983. Citado na página 39.

SALVATORI, R. T.; VENTURA, C. A. A. A agência nacional de saúde suplementar-ans: onze anos de regulação dos planos de saúde. *Organizações & Sociedade*, Universidade Federal da Bahia, v. 19, n. 62, 2012. Citado na página 1.

Saripalli, P.; Tirumala, V.; Chimmad, A. Assessment of healthcare claims rejection risk using machine learning. In: *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*. [S.l.: s.n.], 2017. p. 1–6. Citado na página 11.

SCHöLKOPF, B.; SMOLA, A. J.; MüLLER, K.-R. Advances in kernel methods. In: SCHöLKOPF, B.; BURGES, C. J. C.; SMOLA, A. J. (Ed.). Cambridge, MA, USA: MIT Press, 1999. cap. Kernel Principal Component Analysis, p. 327–352. ISBN 0-262-19416-3. Disponível em: <http://dl.acm.org/citation.cfm?id=299094.299113>. Citado na página 33.

SILVA, C. C. da. Mineraçao de dados-aplicaçao de técnicas de classificaçao para previsao de conhecimento em um contexto médico. 2011. Citado na página 10.

STONE, J. V. Independent component analysis: an introduction. *Trends in Cognitive Sciences*, v. 6, n. 2, p. 59 – 64, 2002. ISSN 1364-6613. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1364661300018131>. Citado 2 vezes nas páginas 36 e 38.

WANG, Q. *Kernel Principal Component Analysis and its Applications in Face Recognition and Active Shape Models*. 2012. Citado na página 32.

WITTEN, I. H. et al. *Data Mining: Practical machine learning tools and techniques*. [S.l.]: Morgan Kaufmann, 2016. Citado 7 vezes nas páginas 10, 19, 20, 21, 27, 44 e 51.

ÜSTüN, B.; MELSSEN, W.; BUYDENS, L. Facilitating the application of support vector regression by using a universal pearson vii function based kernel. *Chemometrics and Intelligent Laboratory Systems*, v. 81, n. 1, p. 29 – 40, 2006. ISSN 0169-7439. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0169743905001474>. Citado na página 32.