UNIVERSIDADE FEDERAL FLUMINENSE

CAROLINA MEDEIROS CARVALHO

UM MODELO DE DECISÃO DINÂMICO APLICADO AO DIAGNÓSTICO DO COMPROMETIMENTO COGNITIVO

NITERÓI

2019

CAROLINA MEDEIROS CARVALHO

UM MODELO DE DECISÃO DINÂMICO APLICADO AO DIAGNÓSTICO DO COMPROMETIMENTO COGNITIVO

Tese apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Doutor. Área de Concentração: Sistemas de Computação.

Orientadora: Profa. Dra. Débora Christina Muchaluat Saade
Coorientadora: Profa. Dra. Aura Conci

Niterói
2019

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

C331m Carvalho, Carolina Medeiros
Um modelo de decisão dinâmico aplicado ao diagnóstico do comprometimento cognitivo / Carolina Medeiros Carvalho ; Débora Christina Muchaluat Saade, orientador ; Aura Conci, coorientador. Niterói, 2019.
150 f. : il.

Tese (doutorado)-Universidade Federal Fluminense, Niterói, 2019.

DOI: http://dx.doi.org/10.22409/PGC.2019.d.09151171775

1. Tomada de decisão clínica. 2. Demência. 3. Comprometimento cognitivo leve. 4. Mineração de dados (Computação). 5. Produção intelectual. I. Saade, Débora Christina Muchaluat, orientador. II. Conci, Aura, coorientador. III. Universidade Federal Fluminense. Instituto de Computação. IV. Título.

CDD -

Bibliotecária responsável: Fabiana Menezes Santos da Silva - CRB7/5274

CAROLINA MEDEIROS CARVALHO

UM MODELO DE DECISÃO DINÂMICO APLICADO AO DIAGNÓSTICO DO COMPROMETIMENTO COGNITIVO

Tese apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Doutor. Área de Concentração: Sistemas de Computação.

Aprovada em abril de 2019.

BANCA EXAMINADORA

Profa. Dra. Débora Christina Muchaluat Saade – Orientadora

UFF

Profa. Dra. Aura Conci - Coorientadora

UFF

Prof. Dr. Alexandre Plastino

UFF

Profa. Dra. Aline Paes

UFF

Profa. Dra. Yolanda Boechat

UFF

Prof. Dr. Antônio Tadeu Azevedo Gomes

LNCC

Prof. Dr. Alexandre Sztajnberg

UERJ

Niterói

2019

Dedico este trabalho a Deus, à minha família e a todos que contribuíram para minha formação acadêmica.

AGRADECIMENTOS

Agradeço aos professores Débora Saade, Aura Conci e Flávio Seixas por todo apoio recebido.

Sou grata ao Dr. Jerson Laks e à Dra. Yolanda Boechat por estarem disponíveis para esclarecer as dúvidas e por disponibilizarem as bases de dados.

Agradeço a meu esposo e filhos pela compreensão nos momentos de ausência.

Meus agradecimentos também ao Instituto Nacional da Propriedade Industrial (INPI), onde trabalho, pelo apoio logístico para cumprir créditos nas disciplinas acadêmicas e elaborar esta tese de doutorado.

“E, demais disto, filho meu, atenta: não há limite para fazer livros, e o muito estudar é enfado da carne. De tudo o que se tem ouvido, o fim é: teme a Deus, e guarda os seus mandamentos; porque isto é o dever de todo o homem. Porque Deus há de trazer a juízo toda a obra, e até tudo o que está encoberto, quer seja bom, quer seja mau.” Eclesiastes 12:12-14.

RESUMO

Objetivando-se a redução dos erros nos diagnósticos, Sistemas de Suporte à Decisão Clínica (CDSS) têm sido utilizados para auxiliar os profissionais da saúde na avaliação e diagnóstico de pacientes. Tais sistemas sugerem, com base nos registros de saúde de certo paciente, um diagnóstico mais provável, o qual pode ser levado em consideração pelo médico ao decidir quanto ao diagnóstico do paciente. Para a sugestão do diagnóstico mais provável, os registros de saúde do paciente são casados com os atributos de um modelo previamente construído utilizando aprendizado de máquina, daqui para frente referenciado como modelo de decisão. Um problema atual é que a maioria dos CDSS está baseada em modelos de decisão estáticos. Nos modelos de decisão estáticos, o aprendizado do modelo é realizado uma única vez, antes do sistema de apoio ao diagnóstico ser posto em operação. Tais modelos estáticos não são refinados ou reconstruídos durante o uso do sistema e, portanto, não permitem que o médico especialista reporte o diagnóstico final para a melhoria do modelo. Na literatura, existem alguns trabalhos baseados em modelos de decisão dinâmicos, mas estes não são adequados à utilização simultânea por centros de saúde distintos que utilizam diferentes conjuntos de exames para diagnosticar a mesma doença, ou seja, utilizam práticas diagnósticas distintas. Nesta tese, propõe-se um modelo de decisão dinâmico e flexível, o qual é alimentado pelos diagnósticos finais reportados por médicos especialistas, sendo adequado a centros de saúde empregando práticas diagnósticas distintas, uma vez que permite que novos exames sejam definidos pelos médicos e o modelo de decisão automaticamente os inclui como atributos, caso sejam potencialmente relevantes para diagnosticar. O modelo de decisão dinâmico proposto baseia-se em um processo de aprendizado de máquina supervisionado que retreina o modelo através da seleção dos atributos considerados potencialmente relevantes, avaliação de vários cenários de pré-processamento e classificadores e seleção da combinação de cenário e classificador que alcança melhor desempenho na classificação considerando um conjunto de medidas. Na aplicação do modelo dinâmico proposto ao diagnóstico do comprometimento cognitivo, um CDSS foi projetado e desenvolvido para auxiliar médicos menos experientes ou especializados a decidirem sobre o diagnóstico de Demência, Doença de Alzheimer (DA) e Comprometimento Cognitivo Leve (CCL). Para cada paciente, o CDSS sugere o diagnóstico mais provável, os registros de saúde mais relevantes que levaram ao diagnóstico sugerido e, no caso de um fator de certeza baixo, o CDSS recomenda testes neuropsicológicos a serem realizados para confirmar ou refutar a sugestão diagnóstica inicial. O CDSS proposto é adequado para centros de saúde que utilizam diferentes conjuntos de

testes neuropsicológicos para diagnosticar Demência, DA e CCL. Dados de pacientes do Centro de Doença de Alzheimer e outras Desordens Mentais na Velhice (CDA) do Instituto de Psiquiatria da Universidade Federal do Rio de Janeiro (UFRJ) e do Centro de Referência em Atenção à Saúde dos Idosos (CRASI) do Hospital Universitário Antonio Pedro da Universidade Federal Fluminense (UFF) foram usados para a realização de testes experimentais. CDA e CRASI aplicam conjuntos distintos não disjuntos de testes neuropsicológicos aos pacientes. Quando os dados do CRASI foram adicionados aos dados do CDA, o classificador final obtido para diagnóstico de CCL melhorou sua generalização com a inclusão de exames exclusivos do CRASI. Além disso, os testes experimentais mostraram que os classificadores obtidos para Demência, DA e CCL alcançaram desempenho competitivo na comparação com outros trabalhos da literatura baseados em modelos dinâmicos.

Palavras-chave: Modelo de Decisão Dinâmico, Sistemas de Suporte à Decisão Clínica, Diagnóstico Auxiliado por Computador, Modelos Preditivos, Demência, Doença de Alzheimer, Comprometimento Cognitivo Leve.

ABSTRACT

Aiming at reducing diagnostic errors, Clinical Decision Support Systems (CDSS) have been used to assist health professionals to evaluate and diagnose patients. Such systems suggest, based on a patient's health records, the most probable diagnosis, which can be taken into consideration by the physician to decide about the patient's diagnosis. To suggest the most probable diagnosis, the patient's health records are matched to the features of a model previously constructed using machine learning, hereafter referenced as decision model. A current problem is that most CDSS are based on static decision models. In static decision models, model learning is performed only once, before the diagnostic support system is put into operation. Such static models are not refined or reconstructed during the use of the system and, therefore, do not allow the expert physician to report the final diagnosis to improve the model. In the literature, there are some works based on dynamic decision models, but these are not suitable for simultaneous use by different health centers that use different sets of tests to diagnose the same disease, that is, use different diagnostic practices. In this thesis, a dynamic and flexible decision model is proposed, which is fed by the final diagnoses reported by expert physicians, being appropriate to health centers that use different diagnostic practices, since it allows new exams to be defined by physicians and the decision model automatically includes them as attributes if they are potentially relevant for diagnosis. The proposed dynamic decision model is based on a supervised machine learning process that retrains the model by selection of attributes considered potentially relevant, assessment of several pre-processing scenarios and classifiers, and selection of the combination of scenario and classifier that achieves better classification performance considering a set of measures. In the application of the proposed dynamic model to diagnose cognitive impairment, a CDSS was designed and implemented to assist less experienced or specialized physicians to decide about diagnosis of Dementia, Alzheimer's Disease (AD) and Mild Cognitive Impairment (MCI). For each patient, the CDSS suggests the most likely diagnosis, the most relevant health records that led to the suggested diagnosis, and in case of a low certainty factor, the CDSS recommends neuropsychological tests to be performed to confirm or refute the initial diagnostic suggestion. The proposed CDSS is suitable for health centers that use different sets of neuropsychological tests to diagnose Dementia, AD and MCI. Patient data from Center for Alzheimer’s Disease (CAD) at Institute of Psychiatry of Federal University of Rio de Janeiro (UFRJ) and from Center of Reference in Attention to Health of the Elderly (CRASI) at Antonio Pedro University Hospital of Fluminense Federal University (UFF) were used to

perform experimental tests. CAD and CRASI apply distinct non-disjoint sets of neuropsychological tests to patients. When CRASI data were added to CAD data, the final classifier obtained for MCI diagnosis improved its generalization with the inclusion of neuropsychological tests that are present only in CRASI data. Besides, the experimental tests showed that the classifiers obtained for Dementia, AD and MCI achieved a competitive performance in comparison to other works of literature based on dynamic models.

Keywords: Dynamic Decision Model, Clinical Decision Support Systems, Computer-aided Diagnosis, Predictive Models, Dementia, Alzheimer's Disease, Mild Cognitive Impairment.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

Sigla	**Termo (Original)**
1H MRS	*Proton Magnetic Resonance Spectroscopy*
A1DE	*Averaged 1-Dependence Estimators*
Acc	Acurácia
AD	*Alzheimer's Disease*
AGG	*Aggressive Brain tumors*
AIVD	Atividades Instrumentais da Vida Diária
AMS	Aprendizado de Máquina Supervisionado
AODE	*Averaged One-Dependence Estimators*
API	*Application Programming Interface*
AUC	*Area under ROC (Receiver Operating Characteristic) Curve*
AVD	Atividades Básicas da Vida Diária
BGL	*Blood Glucose Level*
CBF	*Cerebral Blood Flow*
CCL	Comprometimento Cognitivo Leve
CCL-c	Pacientes com CCL que convertem para DA em até 18 meses
CCL-nc	Pacientes com CCL que não convertem para DA em até 18 meses
CDA	Centro de Doença de Alzheimer e outras Desordens Mentais na Velhice
CDR	*Clinical Dementia Rating*
CDSS	*Clinical Decision Support System*
CDT	*Clock Drawing Test*
CERAD	*Consortium to Establish a Registry for Alzheimer's Disease*
CPD	*Conditional Probability Distribution*
CRASI	Centro de Referência em Atenção à Saúde dos Idosos
CSS	*Cascading Style Sheets*
CV	*Cross-Validation*
DA	Doença de Alzheimer
DAG	*Directed Acyclic Graph*
DER	Diagrama Entidade-Relacionamento
DRA	*Diagnostic Recommendation Algorithm*
DWT	*Discrete Wavelet Transform*
ECG	Eletrocardiograma
EEG	Eletroencefalografia
EM	*Expectation-Maximization algorithm*
FC	Fator de Certeza

FDP	Função de Densidade de Probabilidade
fMRI	*functional Magnetic Resonance Images*
FN	*False Negative*
FP	*False Positive*
FPR	*False Positive Rate*
GDS	*Geriatric Depression Scale*
HTTP	*Hypertext Transfer Protocol,*
iDBLR	*Incremental Discriminative Bayesian Logistic Regression*
iGDA	*Incremental Gaussian Discriminant Analysis*
iLDA	*Incremental Linear Discriminant Analysis*
iPCA	*Incremental Principal Component Analysis*
IQCODE	*Informant Questionnaire on Cognitive Decline in the Elderly*
iOVFDT	*Incrementally Optimized Very Fast Decision Tree*
J2EE	*Java Enterprise Edition*
JPA	*Java Persistence API*
JS	*Java Script*
JSF	*Java Server Faces*
k-NN	*k Nearest Neighbors*
LGG	*Low-Grade Glial tumors*
MCI	*Mild Cognitive Impairment*
MEEM	Mini Exame do Estado Mental
MEN	Meningioma
MLE	*Maximum Likelihood Estimation*
MMH	*Maximum Marginal Hyperplane*
MMSE	*Mini-Mental State Examination*
MVC	*Model-View-Controller*
NB	*Naïve Bayes*
OMS	Organização Mundial da Saúde
PCA	*Principal Component Analysis*
PPV	*Positive Predictive Value*
QP	*Quadratic Programming*
RB	Rede Bayesiana
RBF	*Radial Basis Function*
RF	*Random Forest*
RL	Regressão Logística
RM	Ressonância Magnética

RMSE	*Root Mean Squared Error*
RNC	Rede Neural Convolucional
ROC	*Receiver Operating Characteristic*
ROIs	*Regions Of Interest*
SGBD	Sistema Gerenciador de Banco de Dados
SMO	*Sequential Minimal Optimization*
sm-SVM	*Soft margin SVM*
SMILE	*Structural Modeling, Inference and Learning Engine*
SMOTE	*Synthetic Minority Over-sampling Technique*
SPECT	*Single Photon Emission Computed Tomography*
STROOP	*Stroop color-word naming test*
SVM	*Support Vector Machines*
TC	Tomografia Computadorizada
TCA	Teste Computadorizado de Atenção visual
TCA-E	TCA percentual de Erros
TCA-O	TCA percentual de Omissões
TCA-TR	TCA Tempo de Reação
TCA-VAR	TCA Variabilidade do tempo de reação
TCL	Transtorno Cognitivo Leve
TFN	Taxa de Falsos Negativos
TFP	Taxa de Falsos Positivos
TL	Teste Laboratorial
TMT	*Trail Making Test*
TN	*True Negative*
TNP	Teste Neuropsicológico
TP	*True Positive*
TPR	*True Positive Rate*
UFF	Universidade Federal Fluminense
UFRJ	Universidade Federal do Rio de Janeiro
VFDT	*Very Fast Decision Tree*
VFT	*Verbal Fluency Test*
WHO	*World Health Organization*

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO

Atualmente, na área médica, há uma preocupação crescente quanto à segurança dos pacientes e um esforço para a prevenção e redução dos erros nos diagnósticos (NEWMAN-TOKER e PRONOVOST, 2009). Erros no diagnóstico são definidos como aqueles em que o diagnóstico clínico ocorre de forma tardia ou incorreta a partir da avaliação de informações disponíveis e suficientes do paciente (GRABER, FRANKLIN e GORDON, 2005).

No contexto da redução dos erros diagnósticos, ganham destaque os sistemas computacionais de auxílio ao diagnóstico, principalmente os sistemas de apoio à decisão clínica, conhecidos pela sigla CDSS (*Clinical Decision Support Systems*). Os CDSS são sistemas computacionais utilizados para ajudar os médicos na tomada de decisão no momento do atendimento médico, pertencendo a uma importante categoria dos sistemas de informação em saúde (BERNER, 2007). Projetados para integrar informações clínicas dos pacientes, os CDSS são comumente utilizados para ajudar na decisão médica sobre o diagnóstico ou sobre a terapia a adotar. Estudos demonstraram que os CDSS podem reduzir erros nos diagnósticos médicos (HAYNES e WILCZYNSKI, 2010). Comumente, em tais sistemas, dados de saúde de certo paciente sendo avaliado são casados com dados correspondentes em um modelo previamente construído utilizando aprendizagem de máquina (modelo de apoio à decisão, daqui em diante referenciado como modelo de decisão) para gerar avaliações e recomendações específicas para este paciente, por exemplo, sugerir um diagnóstico mais provável. Os CDSS não visam substituir médicos especialistas, mas podem ajudar com uma segunda opinião. No caso de médicos não tão especializados ou pouco experientes, tais sistemas podem contribuir para a triagem de pacientes possivelmente doentes, fornecendo uma indicação que subsidie o correto encaminhamento do paciente para a realização de exames complementares ou para um médico mais especializado ou experiente. Desta maneira, os CDSS podem contribuir para a antecipação e correta aplicação do tratamento.

Um problema existente na maioria dos sistemas computacionais de auxílio ao diagnóstico da literatura (veja, por exemplo, ILLÁN et al., 2012; SEGOVIA et al., 2013; GAO e HUI, 2016; PAYAN e MONTANA, 2015; KRASHENYI et al., 2015; HU et al., 2016; HOSSEINI-ASL, KEYNTO E EL-BAZ, 2016; SARRAF e TOFIGHI, 2016; SOUNTHARRAJAN e THANGARAJ, 2016; FAROOQ et al., 2017; REZAEI, YANG e MEINEL, 2017; LIU et al., 2015; SEIXAS et al., 2014; WEAKLEY et al., 2015; PEREIRA et al., 2017; BATTISTA, SALVATORE e CASTIGLIONI, 2017; CARVALHO et al., 2017a;

PEKKALA et al., 2017; MORABITO et al., 2016; MIRHEIDARI et al., 2017; COSTA et al., 2016; SEGOVIA et al., 2014; ZHOU et al., 2014) reside no fato destes serem baseados em modelos de decisão estáticos. Modelos de decisão estáticos são modelos de apoio à decisão em que a aprendizagem é realizada uma única vez, antes do sistema de ser posto em operação, ou seja, são modelos de decisão que não são atualizados ou reconstruídos durante o uso do sistema. Portanto, os modelos de decisão estáticos não permitem o aproveitamento dos diagnósticos fornecidos por médicos especialistas para melhorar o modelo e, por conseguinte, não permitem o aperfeiçoamento das sugestões diagnósticas oferecidas pelos sistemas.

Por outro lado, um modelo de decisão é dito dinâmico quando dados rotulados (com diagnóstico confirmado) fornecidos durante a utilização do CDSS são utilizados para atualizar ou reconstruir o modelo. Nos modelos de decisão dinâmicos, diagnósticos confirmados por médicos podem ser usados para melhorar as sugestões diagnósticas providas pelo CDSS promovendo um ciclo contínuo de melhoria.

É de conhecimento geral que, na medicina, os critérios de diagnóstico estão em constante evolução para incluir melhores maneiras de se diagnosticar na medida em que o conhecimento médico avança. Além disso, embora exista um esforço internacional para estabelecer critérios de diagnósticos amplamente aceitos, na prática, podem existir diferenças no conjunto de exames aplicados por centros de saúde distintos para diagnosticar a mesma doença. Sendo assim, é desejável que os CDSS sejam baseados em modelos de decisão que, além de dinâmicos, se ajustem a critérios de diagnóstico que evoluam e a diferenças nos conjuntos de exames aplicados por centros de saúde distintos.

Contudo, os poucos trabalhos da literatura baseados em modelos dinâmicos (ZHANG et al., 2012; FONG et al., 2013; SERHANI, BENHARREF e NUJUM, 2014; SONG et al., 2016; AHMAD, KHAN e MAJEED, 2014; CHEN et al., 2016; ROSADO et al., 2015; VELERT, 2012; CHAN et al., 1993; HUANG e CHEN, 2015; CHO et al., 2012; LEE et al., 2014) não são adequados a critérios de diagnóstico que evoluem, nem permitem a utilização simultânea por centros de saúde com diferentes práticas diagnósticas, pois não permitem a inclusão de atributos potencialmente relevantes durante o uso do sistema.

Além dos erros nos diagnósticos, outro desafio enfrentado pela comunidade médica e científica advém do envelhecimento da população em âmbito mundial, que traz sérios impactos econômicos e sociais. Segundo a Organização Mundial da Saúde (OMS) (WHO, 2013), estima-se que o número de pessoas com mais de 60 anos alcançará a proporção de 22% da população mundial em 2050, em torno de dois bilhões de pessoas. Esta longevidade

populacional leva, consequentemente, ao aumento da prevalência das chamadas doenças da terceira idade, nas quais o diagnóstico precoce e correto é de suma importância para o aumento na expectativa e qualidade de vida. Dentre as doenças com alta prevalência na terceira idade, destacam-se os transtornos mentais e neurológicos (mais de 20% entre os idosos, segundo a OMS) dentre os quais se destacam a Doença de Alzheimer (DA) e transtornos relacionados.

A DA é uma doença neurodegenerativa, progressiva, crônica e irreversível causando lesões no encéfalo, mas é possível fazê-la progredir mais lentamente através do tratamento adequado e da detecção em estágios iniciais. Inicialmente, a DA afeta a memória de curto prazo e, na medida em que a doença avança, as deficiências cognitivas se agravam causando dificuldade de leitura e afetando a percepção espacial, o reconhecimento de faces e objetos, as funções executivas e a solução de problemas. Sintomas em estágios avançados incluem julgamento prejudicado, desorientação, confusão, mudanças comportamentais e dificuldade para andar, falar e até mesmo deglutir. Segundo a Associação Americana de Alzheimer (ALZHEIMER'S A., 2017), a DA é considerada como a sexta causa de morte entre grupos de todas as idades nos Estados Unidos. Entre 2000 e 2014, mortes resultantes de derrame, doenças cardíacas e câncer de próstata diminuíram 21%, 14% e 9%, respectivamente, enquanto mortes decorrentes de DA aumentaram em 89%. Além disso, essa Associação estima que, em 2050, o número de pessoas com DA vivendo nos Estados Unidos seja de 13,8 milhões.

A DA é considerada o tipo mais comum de Demência, sendo responsável por 60% a 80% dos casos de Demência (SOSA-ORTIZ et al., 2012). Outros tipos de Demência incluem a Demência vascular, a Demência frontotemporal e a Demência com corpos de Lewy (CARAMELLI e BARBOSA, 2002). A Associação Americana de Alzheimer (ALZHEIMER'S A, 2017) afirma que, nos Estados Unidos, as despesas totais com assistência à saúde, cuidados de longo prazo e serviços hospitalares para pessoas com 65 anos ou mais acometidas por Demência, somaram 259 bilhões de dólares em 2017. A prevalência de Demência na América Latina é similar a dos países desenvolvidos (NITRINI et al., 2009). Um estudo epidemiológico com a população brasileira estimou que 8% da população idosa tenha Demência (BOTTINO et al., 2008).

Um transtorno importante relacionado à DA é o Comprometimento Cognitivo Leve (CCL) [1], que usualmente está associado a um estágio pré-clínico de DA. Déficit na memória episódica (habilidade de recordar eventos específicos relativos a um dado momento e lugar) é o sintoma mais comum de pacientes com CCL (ALBERT et al., 2011). Um estudo conduzido por CELSIS (2000) revelou que, anualmente, de 10 a 30% dos pacientes que recebem um diagnóstico positivo para CCL convertem para DA, enquanto a taxa de conversão entre idosos normais varia entre 1 e 2%. Portanto, o diagnóstico precoce e correto para CCL pode contribuir para antecipar o tratamento e melhorar a qualidade de vida do paciente.

1.1. OBJETIVOS

Diante da necessidade de um modelo de decisão adequado a critérios de diagnóstico e conjuntos de dados que evoluem ao longo do tempo, o principal objetivo desta tese é propor um modelo de decisão dinâmico que torne o CDSS onde for aplicado flexível para ser utilizado por centros de saúde distintos e que aprenda com os novos dados rotulados fornecidos pelos médicos que utilizem o CDSS em suas rotinas clínicas. Portanto, o modelo proposto permite a inclusão automática de exames potencialmente relevantes como atributos. Para isto, os sistemas implementando o modelo proposto devem ser capazes de permitir a personalização com a definição de novos tipos de exames, por exemplo, testes neuropsicológicos (TNPs) comumente utilizados no centro de saúde e os diagnósticos providos pelos médicos especialistas devem ser usados para alimentar o conjunto de dados de treinamento respectivo para a doença a diagnosticar. O modelo de decisão dinâmico é construído por um método computacional proposto, baseado em Aprendizado de Máquina Supervisionado (AMS), que leva em consideração os exames recém-definidos. Tal método avalia vários classificadores e cenários de pré-processamento e seleciona automaticamente a combinação de classificador e cenário que alcança melhor desempenho na classificação, considerando-se diferentes medidas na avaliação. Este método de aprendizado automaticamente incorpora ao modelo de decisão os exames recém-definidos, desde que estes tenham sido frequentemente aplicados na rotina clínica das unidades de saúde em questão, possuindo pontuações suficientemente informadas no sistema, e apresentem ganhos de informação não desprezíveis.

[1] Comprometimento Cognitivo Leve (CCL) também é chamado de Transtorno Cognitivo Leve (TCL) por alguns autores.

Além disso, diante da incidência crescente das doenças da terceira idade, particularmente dos transtornos que comprometem a cognição, e da necessidade de auxiliar os profissionais de saúde na tomada de decisão no processo diagnóstico de tais transtornos visando à redução dos erros nos diagnósticos, outro objetivo desta tese é aplicar o modelo dinâmico proposto a um CDSS, projetado e desenvolvido para ser facilmente acessado a partir de dispositivos móveis para auxiliar no diagnóstico de Demência, DA e CCL. O sistema terá um impacto potencialmente significativo para a melhoria na qualidade de vida dos pacientes ao contribuir para que se inicie o tratamento em estágios iniciais do comprometimento cognitivo. Ainda, o CDSS desenvolvido objetiva o alinhamento à medicina baseada em evidências permitindo a identificação dos critérios que levaram ao diagnóstico sugerido através de um paradigma quantitativo.

Outro objetivo da tese é testar a aplicação do modelo dinâmico a bases de dados clínicos de pacientes reais de dois centros de saúde. Tais bases de dados contêm diagnósticos confirmados por médicos especialistas para pacientes avaliados cognitivamente para a detecção de Demência, DA e CCL.

1.2. CONTRIBUIÇÕES DA TESE

As principais contribuições desta tese de doutorado são:

(1) Um modelo de decisão dinâmico que se ajusta a novos dados clínicos com diagnóstico confirmado inseridos no sistema além de permitir a inclusão de novos atributos. O modelo de decisão é flexível, pois pode ser utilizado por centros de saúde utilizando diferentes conjuntos de exames, pois o método de aprendizado que o constrói automaticamente leva em consideração exames potencialmente relevantes para o diagnóstico de acordo com os dados dos pacientes atualizados na base de dados do sistema por profissionais destes centros de saúde.

(2) A aplicação do modelo de decisão dinâmico proposto ao diagnóstico de Demência, DA e CCL. Para isto, foi projetado e desenvolvido um CDSS que implementa o modelo de decisão dinâmico, contendo três classificadores binários respectivos para diagnóstico destas três doenças. O CDSS é capaz de predizer o diagnóstico mais provável (negativo ou positivo), baseando-se em atributos que incluem fatores predisponentes e pontuações obtidas na aplicação de TNPs. Além disso, o CDSS foi projetado para permitir o acesso a partir de dispositivos móveis, o que torna propícia a sua utilização por profissionais da saúde e facilita que pacientes

com dificuldades de locomoção sejam atendidos em suas próprias casas ou em lugares mais convenientes.

O modelo de decisão dinâmico proposto por esta tese pode ser implementado por outros sistemas de apoio diagnóstico desde que, em tais sistemas, novos atributos possam ser definidos usando a interface de usuário do sistema de acordo com necessidade das práticas diagnósticas dos centros de saúde e médicos especialistas alimentem o sistema com suas decisões diagnósticas. Além disso, o CDSS desenvolvido possui uma arquitetura baseada em componentes e utiliza padrões abertos e bem documentados refletindo boas práticas da programação orientada a objetos. Isso permite o reuso de componentes deste CDSS por outros sistemas médicos de apoio diagnóstico. Portanto, tanto o modelo dinâmico como o CDSS proveem bases para futuros avanços nos cuidados à saúde.

Como dito anteriormente, na construção do modelo dinâmico, um método de aprendizado supervisionado avalia vários classificadores e cenários de pré-processamento e seleciona automaticamente a combinação de classificador e cenário que alcança melhor desempenho na classificação, considerando-se diferentes medidas na avaliação. Outra contribuição desta tese é a implementação de dois algoritmos de classificação que são avaliados e podem ser escolhidos por este método de aprendizado:

- Uma Rede Bayesiana (RB ou *Bayesian Network*, BN) discreta com estrutura fornecida, composta por três níveis (o primeiro nível contendo nós representando fatores predisponentes, o segundo nível contendo o nó representando a doença a ser diagnosticada e o terceiro nível contendo nós representando resultados de exames/testes aplicados sobre os pacientes), automatizando a modelagem e o aprendizado de parâmetros propostos por SEIXAS et al. (2014).
- Um modelo híbrido combinando Regressão Logística e Naïve Bayes (*Hybrid Logistic Regression-Naïve Bayes Model*), como descrito por CARVALHO et al. (2017b), equivalente a uma RB com estrutura fornecida composta pelos três níveis acima mencionados, porém capaz de combinar nós discretos e contínuos.

Este trabalho foi desenvolvido no âmbito do Projeto SIADE - Pesquisas em Sistemas de Apoio à Decisão e ao Diagnóstico de Doenças Associadas ao Envelhecimento, aprovado pelo Comitê de Ética em Pesquisa CEP/CONEP em 05/10/2015.

1.3. ESTRUTURA DO TEXTO

O Capítulo 2 descreve trabalhos relacionados a modelos de decisão estáticos (com foco em DA e transtornos relacionados) ou a modelos de decisão dinâmicos utilizados por sistemas de auxílio ao diagnóstico de diversas doenças.

O Capítulo 3 fornece fundamentação teórica quanto à tarefa de classificação em geral e quanto aos classificadores que podem ser escolhidos pelo método de aprendizado que constrói o modelo de decisão dinâmico proposto.

O Capítulo 4 descreve o modelo de decisão dinâmico, o método de aprendizado que o constrói e como este modelo pode ser utilizado para indicar exames mais relevantes para um diagnóstico sugerido ou recomendar exames futuros.

O Capítulo 5 aborda a aplicação do modelo de decisão dinâmico a um CDSS que apoia o diagnóstico de Demência, DA e CCL, descrevendo seus principais componentes, arquitetura e apoio à decisão provido.

O Capítulo 6 detalha as bases de dados utilizadas em testes experimentais realizados, além de apresentar e discutir os resultados de tais testes.

Por fim, o Capítulo 7 realça as principais contribuições da tese e apresenta sugestões para trabalhos futuros.

CAPÍTULO 2 – TRABALHOS RELACIONADOS

Este capítulo tem como objetivo fazer uma breve exposição sobre trabalhos relacionados à tese. Os trabalhos relacionados foram divididos em dois conjuntos distintos: a Seção 2.1 aborda trabalhos que propõem modelos de decisão estáticos para o diagnóstico de DA e transtornos relacionados e a Seção 2.2 descreve trabalhos para auxílio ao diagnóstico de diversas doenças baseados em modelos de decisão dinâmicos.

2.1 MODELOS DE DECISÃO ESTÁTICOS PARA AUXÍLIO AO DIAGNÓSTICO DE DA E TRANSTORNOS RELACIONADOS

Existem vários trabalhos para o auxílio computacional ao diagnóstico de DA e transtornos relacionados, mas a maioria deles baseia-se em modelos de decisão estáticos.

Alguns desses trabalhos construíram seus modelos de decisão estáticos com base em análise de imagens, incluindo:

- Tomografia Computadorizada (TC) (SALAS-GONZALEZ et al., 2010; ILLÁN et al., 2011; ILLÁN et al., 2012; CHAVES et al., 2013; SEGOVIA et al., 2013; GAO e HUI, 2016).
- Ressonância Magnética (RM) (SINGH e CHETTY, 2012; ORTIZ et al., 2013; LIU et al., 2014; PAYAN e MONTANA, 2015; KRASHENYI et al., 2015; HU et al., 2016; HOSSEINI-ASL, KEYNTO E EL-BAZ, 2016; SARRAF e TOFIGHI, 2016; SOUNTHARRAJAN e THANGARAJ, 2016; FAROOQ et al., 2017; REZAEI, YANG e MEINEL, 2017).
- Análise multimodal de imagem (SUK e SHEN, 2013; LIU et al., 2015; LI et al., 2015; BHATKOTI e PAUL, 2016).

Outros trabalhos construíram seus modelos estáticos baseando-se em dados clínicos incluindo TNPs e fatores predisponentes (GARCÍA et al., 2012; SEIXAS et al., 2014; WEAKLEY et al., 2015; MOREIRA e NAMEN, 2016; GUERRERO et al., 2016; PEREIRA et al., 2017; BATTISTA, SALVATORE e CASTIGLIONI, 2017; CARVALHO et al., 2017a; PEKKALA et al., 2017). Outros trabalhos usaram sinais de eletroencefalogramas (EEGs) (TRAMBAIOLLI et al., 2011; MORABITO et al., 2016) ou análise de conversação (MIRHEIDARI et al., 2017) para diagnosticar DA e transtornos relacionados.

Existem também trabalhos combinando diferentes tipos de dados:

- Dados coletados por sensores combinados com dados demográficos e resultados de TNPs (COSTA et al., 2016);

- Análise de imagens combinada com dados demográficos e dados clínicos coletados durante a aquisição das imagens (TRIPOLITI, FOTIADIS e ARGYROPOULOU, 2011);
- Análise de imagens combinada com dados demográficos e resultados de TNPs (SUN, LV e TANG, 2007; SEGOVIA et al., 2014; ZHOU et al., 2014);
- Análise de imagens combinada com resultados de testes laboratoriais (TLs) e TNPs (MATTILA et al., 2012).

2.2 MODELOS DE DECISÃO DINÂMICOS PARA AUXÍLIO AO DIAGNÓSTICO DE DIVERSAS DOENÇAS

Alguns dos trabalhos baseados em modelos dinâmicos são baseados em *streams* de dados (ZHANG et al., 2012; FONG et al., 2013; SERHANI, BENHARREF e NUJUM, 2014), em que dados chegam a uma taxa muito alta de forma que seu armazenamento nem sempre é possível. Tais sistemas usam o paradigma de aprendizado incremental, no qual, cada lote contendo novos dados rotulados (novos registros com diagnóstico confirmado) é usado para refinar um modelo aprendido anteriormente, assumindo a premissa de que os dados utilizados nos aprendizados anteriores não estão mais disponíveis, apenas o último modelo aprendido. Os sistemas baseados em *streams* podem empregar o aprendizado incremental do tipo on-line, em que o modelo é atualizado a cada novo registro de treinamento.

ZHANG et al. (2012) propuseram uma aplicação de software genérica que pode ser usada no prognóstico e diagnóstico de doenças crônicas como a diabetes. Os atributos utilizados incluem dados de pressão sanguínea, temperatura corporal, e *features* extraídas de eletrocardiogramas (ECGs) e de EEGs. Conjuntos de dados em um determinado instante de tempo formam um vetor de medidas coletadas por sensores. A aplicação baseia-se em um algoritmo de classificação denominado Árvore de Decisão Muito Rápida (*Very Fast Decision Tree* (VFDT), DOMINGOS e HULTEN, 2000). Um novo registro contendo diagnóstico e a árvore antiga são usados para treinar uma nova árvore.

FONG et al. (2013) propuseram um CDSS de tempo real para auxiliar na terapia de pacientes com diabetes através da detecção de condições normais e anormais quanto ao nível de glicose no sangue (*Blood Glucose Level* - BGL). O objetivo é predizer um BGL futuro, dados os seguintes eventos: injeções de insulina, tipo de refeições ingeridas, execução de exercícios físicos e níveis históricos de BGL. O número de classes é sete, representando diferentes BGLs. A construção do modelo inicial tem como entrada uma pequena porção dos dados iniciais e, posteriormente, o classificador aprende e prediz simultaneamente. Quatro

algoritmos de classificação com hiperparâmetros fixos foram avaliados: VFDT (DOMINGOS e HULTEN, 2000), iOVFDT (YANG e FONG, 2013), *Bayes* (ANWAR, ASGHAR e FONG, 2011) e redes neurais artificiais modeladas por ROBERTSON et al. (2011). Porém, o classificador do CDSS não é dinamicamente escolhido. Como método de avaliação, cada registro foi dividido em duas partes: uma representando dados médicos históricos usados para o treinamento e a outra parte representando dados médicos futuros para teste. Para cada algoritmo de classificação, todos os registros foram testados e os desempenhos foram medidos considerando uma janela deslizante de 24 horas para o treinamento. Cada medida de desempenho foi calculada como uma média dos respectivos desempenhos obtidos considerando as predições ao longo de todo o período da terapia.

SERHANI, BENHARREF e NUJUM (2014) descreveram um CDSS que suporta monitoramento móvel contínuo processando leituras e aplicando um conjunto de regras para gerar assistência automatizada a pacientes em monitoramento. As assistências geradas são visualizadas por usuários em seus próprios dispositivos móveis e por médicos através de um site na Internet. O sistema auxilia no diagnóstico de onze doenças cardíacas diferenciando-as de uma condição considerada normal. *Features* são extraídas de ECGs coletados em tempo real e outros atributos incluem medidas de temperatura, pressão arterial e batimentos cardíacos coletadas por sensores. O modelo de decisão é dinâmico: um processo inteligente continuamente é executado e consiste em mineração incremental dos dados coletados. Regras são extraídas usando conhecimento adquirido através de aprendizado e dos dados coletados continuamente e são enriquecidas com novos conjuntos de regras. Os seguintes classificadores foram avaliados, mas o algoritmo de classificação não é dinamicamente escolhido: redes neurais, árvore de decisão, AutoMLP, W-OneR, Naïve Bayes (NB), RuleModel e J48graft[2]. O desempenho foi avaliado em termos do número de classes que o sistema consegue detectar e a acurácia (Acc) na detecção de cada classe contra as outras classes foi determinada para cada classificador.

Alguns autores (AHMAD, KHAN e MAJEED, 2014; CHEN et al., 2016; ROSADO et al., 2015) propuseram retreinamento para proporcionar modelos de decisão dinâmicos. Nos modelos dinâmicos baseados em retreinamento, os dados rotulados são armazenados na base na medida em que são disponibilizados e o aprendizado utiliza todos os dados armazenados para reconstruir o modelo.

[2] Para saber mais sobre estes classificadores, consulte a documentação da ferramenta Rapidminer: https://docs.rapidminer.com/

AHMAD, KHAN e MAJEED (2014) propuseram um sistema para predizer epilepsia baseando-se em EEGs. O modelo é dinâmico: se o médico discorda com a classificação predita pelo sistema, ele informa a classificação corrigida e esta informação é levada em consideração quando o retreinamento é acionado por um usuário. O CDSS também permite a criação de modelos individuais por usuário *logado*. Sinais representando cada época (pedaço do sinal com duração de 1s) por canal para os tipos de padrão epilético são processados usando técnicas de processamento de sinais e aprendizado supervisionado. Transformada Discreta de *Wavelet* (*Discrete Wavelet Transform* - DWT) é aplicada sobre cada época e, então, *features* dos sinais (a média μ, o desvio padrão σ e a potência) são calculadas a partir de coeficientes selecionados. Tais *features* são então reduzidas usando Análise de Componentes Principais (*Principal Component Analysis* - PCA) e alimentam um classificador *Support Vector Machines* (SVM) com núcleo (*kernel*) linear para classificar uma época como epilética ou não. Como método de avaliação, validação cruzada (C*ross-Validation* - CV) com 10 *folds* foi aplicada.

CHEN et al. (2016) também descreveram um sistema que prediz epilepsia baseando-se em análise de EEGs, Transformada *Wavelet* e SVM. Durante o treinamento, EEGs epiléticos e normais, que foram marcados por especialistas, têm seus coeficientes extraídos para as bandas compreendidas entre 1 e 32, sendo os coeficientes das demais bandas igualados a zero. Então, um modelo é treinado usando SVM com núcleo *Radial Basis Function* (RBF) e os valores para gama (hiperparâmetro da função RBF) e para C (hiperparâmetro que atua sobre a regularização para evitar *overfitting*[3]), são obtidos através de CV. Durante a predição, EEGs de uma pessoa são automaticamente decompostos e usados para predizer se tal pessoa tem uma onda característica para epilepsia usando o modelo previamente construído. Se a pessoa possui um sinal predito como anormal, isto é, epilético, seus dados são usados para retreinamento do modelo, desta maneira provendo um sistema dinâmico. Portanto, o algoritmo de classificação é fixo (SVM) com hiperparâmetros sintonizados através de CV. Como método de avaliação, dados de treinamento foram progressivamente aumentados variando de 10% a 90% do conjunto total de treinamento em incrementos de 10%. Para cada percentual usado no treinamento, a Acc foi determinada e plotada usando CV e dados de teste em separado.

[3] *Overfitting*, ou sobreajuste, ocorre quando um classificador se ajusta tanto aos exemplos de treinamento (dentro da amostra) que isto impacta negativamente o desempenho do classificador para dados não utilizados no treinamento (fora da amostra) implicando que o erro dentro da amostra é muito menor que o erro fora da amostra. Para evitá-lo, procura-se não usar modelos demasiadamente complexos principalmente para bases de treinamento com número pequeno de tuplas.

ROSADO et al. (2015) descreveram um sistema para predizer o risco de uma lesão de pele ser um câncer. Usando uma aplicação móvel, pacientes podem enviar um *check-up* para um servidor, contendo uma foto da lesão e informações adicionais como a parte do corpo em que a lesão está localizada. As pontuações de critérios quanto à assimetria, borda e cor da lesão são extraídas por processamento de imagem e o risco da lesão ser um câncer é, então, determinado. O sistema gera um relatório para cada *check-up*, que auxiliará os médicos na avaliação do risco. Depois da validação do médico, um relatório é enviado para a aplicação móvel e os dados validados são usados para retreinar o classificador quando um usuário administrador ativa o retreinamento.

Alguns trabalhos propuseram modelos de decisão dinâmicos baseados em aprendizado incremental para auxílio ao diagnóstico de DA e transtornos relacionados (CHAN et al., 1993; HUANG e CHEN, 2015; CHO et al., 2012; e LEE et al., 2014).

CHAN et al. (1993) propuseram um sistema para predizer DA baseando-se em medidas de fluxo sanguíneo cerebral (*Cerebral Blood Flow* - CBF) de 16 *Regions Of Interest* (ROIs) a partir de imagens *Single Photon Emission Computed Tomography* (SPECT). O algoritmo de classificação proposto foi rede neural artificial incremental. Uma rede neural inicial é treinada antes do sistema ser posto em operação e, posteriormente, adaptada com a adição de neurônios e pesos quando novas amostras rotuladas são providas durante a operação do sistema quando são fornecidos diagnósticos feitos por médicos especialistas. Como método de avaliação, *leave-one-out* foi utilizado.

HUANG e CHEN (2015) propuseram um sistema incremental para predizer DA e CCL baseando-se em medidas de CBF de ROIs de imagens de ressonância magnética funcional fMRI (*functional Magnetic Resonance Images*) e em um algoritmo conhecido por clusterização espectral em sua forma incremental. O método de avaliação utilizado foi CV com 5 *folds*.

CHO et al. (2012) descreveram um sistema que classifica pacientes em: pacientes normais, com DA, CCL conversores (CCL-c) e CCL não conversores (CCL-nc). CCL-c são definidos como pacientes com CCL que se convertem para pacientes com DA nos próximos 18 meses e CCL-nc permanecem estáveis ou mesmo revertem para uma condição normal no mesmo período. As *features* consistem em dados de espessura cortical (*cortical thickness*) dos pacientes em termos dos componentes de frequência extraídos de RM e o algoritmo de classificação utilizado é PCA incremental (iPCA) (HALL, MARSHALL e MARTIN, 1998) combinado com análise de discriminante linear incremental (*Incremental Linear Discriminant Analysis* - iLDA) (PANG, OZAWA e KASABOV, 2005). O método de avaliação utilizado

foi subamostragem aleatória (*random subsampling*) com divisão aleatória da base total em treinamento (50%) e teste (50%) executada 100 vezes. Em cada execução, foi realizada a avaliação a cada lote de 10 tuplas da base de treinamento de forma incremental, obtendo-se o desempenho a cada lote. Então, a média das medidas de desempenho para todas as execuções foi obtida para cada lote e colocada em um gráfico.

LEE et al. (2014) realizaram a predição de DA baseando-se em dados de espessura cortical e formato do hipocampo extraídos de RM e também utilizaram iPCA combinado com iLDA. O treinamento inicial utilizou 20% da base e Acc, sensibilidade e especificidade foram medidas a cada incremento de 10% da base, mas o artigo não deixa claro como treinamento e teste são separados.

Outros trabalhos baseados em modelos dinâmicos para auxílio ao diagnóstico (SONG et al., 2016; VELERT, 2012) não empregam *streams* de dados ou retreinamento, nem são direcionados ao diagnóstico de DA e transtornos relacionados. Tais trabalhos utilizam algoritmos próprios para auxiliar no diagnóstico de outras doenças.

SONG et al. (2016) propuseram um sistema dinâmico para rastreio de câncer de mama que visa reduzir o número de biópsias desnecessárias. O sistema diagnostica uma massa mamária como maligna ou benigna exibindo o diagnóstico como uma probabilidade da massa ser maligna. O limiar de decisão é dinâmico, isto é, não é fixo em 0,5, variando para minimizar a Taxa de Falsos Positivos (TFP) sujeito a uma condição de taxa máxima para os falsos negativos. O sistema recomenda a melhor ação para um dado paciente, que pode ser 0 (caso o limite não seja excedido) ou 1 (caso contrário). A ação 0 indica que o paciente deve retornar após um determinado período de tempo para revisão e a ação 1 indica que o paciente deve fazer uma biópsia. Os atributos dos pacientes são denominados pelos autores como informação contextual e compreendem informação demográfica, densidade mamária, informação sobre a mama oposta e a modalidade usada no exame de imagem. Um algoritmo chamado de Algoritmo de Recomendação Diagnóstica (*Diagnostic Recommendation Algorithm* - DRA) acumula os dados e adapta a estratégia diagnóstica (recomendação) ao longo do tempo. A ideia principal é, de maneira adaptativa, agrupar os dados dos pacientes baseando-se nos contextos associados e aprender a melhor ação para cada agrupamento. Os agrupamentos são dinamicamente particionados e não podem conter mais que certo número de pacientes dentro dele. Registros de pacientes dentro do mesmo cluster possuem a mesma probabilidade diagnóstica e ação recomendada. Para um dado agrupamento, tal probabilidade é automaticamente atualizada quando o agrupamento em questão excede o número máximo de pacientes dentro dele ou há uma discordância do médico com relação à ação sugerida pelo

sistema para um paciente dentro deste agrupamento. Portanto, o algoritmo mantém uma estimativa de probabilidade dos pacientes em cada agrupamento possuírem tumor maligno e varia o limiar de decisão de forma a assegurar que a Taxa de Falsos Negativos (TFN) seja menor que um valor predeterminado (por exemplo, 2% ou 5%) e a TFP alcançada seja mínima.

VELERT (2012) descreveu um sistema de classificação de tumores cerebrais baseado em dados de espectroscopia de ressonância magnética de prótons *(Proton Magnetic Resonance Spectroscopy* -1H MRS). Três tipos de tumores são preditos: tumores cerebrais agressivos (*Aggressive Brain tumors* - AGG) que incluem glioblastomas e metástases; tumores gliais de baixa graduação (*Low-Grade Glial tumors* - LGG), que incluem astrocitoma grau II, oligodendroglioma e oligoastrocitoma; e meningioma (MEN). *Features* são extraídas de dados 1H MRS em que padrões espectrais contêm picos relacionados a diferentes concentrações de metabólitos no tecido analisado, os quais são úteis para a classificação dos tumores. Quinze *features* são obtidas a partir da integração do sinal sob regiões espectrais associadas a cada metabólito de interesse. Dois algoritmos de classificação foram propostos e avaliados, porém não são escolhidos automaticamente pelo sistema: (1) Análise Discriminante Gaussiana Incremental (*Incremental Gaussian Discriminant Analysis* - iGDA), baseado na combinação de estimadores de máxima verossimilhança, que assume que os dados seguem uma distribuição Gaussiana multivariada; e (2) Regressão Logística Bayesiana Discriminativa Incremental (*Incremental Discriminative Bayesian Logistic Regression* - iDBLR). Como método de avaliação, a base de dados inteira foi aleatoriamente repartida em uma partição de treinamento com 39,5 % da base (300 tuplas) e uma partição de teste com 60,5% da base (460 tuplas). A partição de treinamento foi dividida em 10 subconjuntos de 30 tuplas para treinamento incremental. A partição inteira de testes foi usada como um conjunto independente para cada novo classificador atualizado. O experimento foi repetido 100 vezes e a Acc média foi indicada considerando os lotes de treinamento.

2.3 COMPARAÇÃO ENTRE A METODOLOGIA PROPOSTA E TRABALHOS RELACIONADOS

A Tabela 1 resume as condições de saúde diagnosticadas com auxílio computacional e as técnicas utilizadas pelos trabalhos da Seção 2.1, que se baseiam em modelos de decisão estáticos. Sendo assim, nenhum dos trabalhos da Tabela 1 permite que a opinião médica diagnóstica seja utilizada para melhorar o modelo de decisão.

O modelo proposto na presente tese difere dos trabalhos da Tabela 1 por ser um

modelo de decisão dinâmico. Já os trabalhos comentados na Seção 2.2 estão mais relacionados à proposta desta tese.

As Tabelas 2 a 5 resumem as principais características dos trabalhos da Seção 2.2, que se baseiam em modelos de decisão dinâmicos, e incluem detalhes sobre os conjuntos de dados utilizados e sobre o desempenho alcançado.

Trabalhos baseados em *streams* de dados (vide Tabela 2) diferem do modelo proposto nesta tese, porque se referem à predição de doenças e condições de saúde que requerem o monitoramento contínuo, isto é, são comumente utilizados em unidades de terapia intensiva e coletam grandes quantidades de dados fisiológicos requerendo a atualização do modelo de decisão quase em tempo real. Por outro lado, o modelo proposto é direcionado a sistemas cujos dados utilizados no processo de aprendizado compreendem registros clínicos que são informados com menor frequência, o que permite o armazenamento na base de dados para o aprendizado futuro. Neste sentido, a presente tese difere dos trabalhos baseados em *streams*, aproximando-se dos trabalhos baseados em retreinamento (vide Tabela 3), mas difere dos trabalhos da Tabela 3 por permitir a inclusão de atributos potencialmente relevantes para o diagnóstico durante o uso do sistema que atualiza a base de casos clínicos.

Outra diferença a ressaltar é que o modelo dinâmico proposto nesta tese não é direcionado a sistemas de auxílio a diagnóstico que fazem a extração automática das *features* como parte do algoritmo que antecede a classificação (típicos dos sistemas baseados em análise de imagem) ou dentro do próprio algoritmo de classificação (que é o caso das Redes Neurais Convolucionais - RNCs). Sendo assim, o modelo dinâmico da tese difere dos modelos de decisão da Tabela 4, por não envolver análise de imagem. Ressalta-se que os conjuntos de dados que foram fornecidos pelos centros de saúde parceiros CDA/UFRJ e CRASI/UFF para alimentar o modelo proposto não incluem dados de imagem dos pacientes, apenas dados clínicos e exames, representados por atributos contínuos ou discretos (também chamados de categóricos). Portanto, o modelo de decisão dinâmico proposto se restringe a sistemas de auxílio a diagnóstico baseados em tipos de dados que possam ser representados por atributos contínuos ou discretos.

O modelo de decisão dinâmico proposto é direcionado a sistemas de apoio ao diagnóstico em que novos atributos possam ser definidos através da interface de usuário e a realimentação diagnóstica de médicos especialistas seja fornecida durante a rotina clínica.

Ressalte-se que nenhum dos trabalhos relacionados baseados em modelos de decisão dinâmicos (vide Tabelas 2 a 5) possui as seguintes características originais desta tese:

(1) Atributos (tipos de exames) podem ser adicionados pelos médicos sendo

automaticamente incorporados ao modelo de decisão se o respectivo exame é aplicado com frequência aos pacientes e apresenta ganho de informação não desprezível.

(2) Para cada doença a ser predita, o algoritmo de classificação e as etapas de pré-processamento podem mudar ao longo do tempo de acordo com o respectivo conjunto de dados de treinamento e tanto as etapas de pré-processamento quanto o algoritmo de classificação são automaticamente escolhidos de acordo com o melhor resultado de desempenho estimado na classificação.

Tabela 1 - Trabalhos relacionados abordando modelos de decisão estáticos para o auxílio computacional ao diagnóstico de DA e transtornos relacionados.

Referência	**Doenças/condições preditas**	**Técnicas utilizadas**
SALAS-GONZALEZ et al. (2010)	DA e Normal (N).	Extração de μ e σ de *voxels* selecionados a partir de SPECT. Algoritmos de classificação avaliados: SVM com núcleo linear e árvore de decisão CART.
ILLÁN et al. (2011)	DA em diferentes estágios de progressão e normal.	Técnica de redução de *features* baseada em máscara e método de votação para a agregação de classificadores SVM.
ILLÁN et al. (2012)	DA em diferentes estágios de progressão e normal.	Extração de componentes relacionados à simetria de estruturas cerebrais a partir de SPECT e classificação usando SVM.
CHAVES et al. (2013)	DA e N.	Extração de μ das intensidades dos *voxels* das ROIs, discretização e classificação por regras de associação.
SEGOVIA et al. (2013)	DA e N.	Algoritmo *Partial Least Squares* para extração de vetores de pontuação a partir de *voxels* selecionados de SPECT, erro *Out-Of-Bag* para seleção dos vetores usados como *features* e classificador SVM.
GAO e HUI (2016)	DA, tumor cerebral e N.	*Deep Learning* (RNCs 2D e 3D).
SINGH e CHETTY (2012)	Cérebro normal e anormal (DA, doença de Parkinson, doença de Huntington e outras doenças que causam lesões cerebrais).	Segmentação de lesões, DWT, seleção de *features*, *Deep Belief Networks* e *Extreme Machine Learning*.
ORTIZ et al. (2013)	DA e N.	Segmentação dos tecidos cerebrais[4] para extração dos volumes, *Learning Vector Quantization* e SVM com núcleo RBF.
LIU et al. (2014)	DA, CCL-c, CCL-nc e N.	Extração do volume do tecido cerebral cinza, *Deep Learning* (*stacked auto-encoders* seguidos de uma camada de saída *softmax*).
PAYAN e MONTANA (2015)	DA, CCL e N.	*Deep Learning* (*sparse auto-encoder* seguido de rede neural convolucional 3D).
KRASHENYI et al.	DA, CCL e N.	Extração de μ e σ das intensidades dos voxels de

[4] Matéria branca (*White Matter*), Matéria Cinza (*Gray Matter*) e Fluido cérebro-espinhal (*Cerebrospinal fluid*).

(2015)		ROIs e sistema de inferência *Fuzzy*.
HU et al. (2016)	CCL e N.	Extração da conectividade funcional entre diferentes regiões do cérebro e *Deep Learning* (*auto-encoder*).
HOSSEINI-ASL, KEYNTO E EL-BAZ (2016)	DA, CCL e N.	*Deep Learning* (a*uto-encoder* convolucional 3D pré-treinado para a extração de *features* seguido de camadas treinadas para a tarefa específica da classificação).
SARRAF e TOFIGHI (2016)	DA e N.	*Deep Learning* (Rede neural convolucional com arquitetura LeNet-5).
SOUNTHARRAJAN e THANGARAJ (2016)	DA, CCL e N.	*Features* são extraídas dos tecidos cerebrais e um algoritmo híbrido *Fish Swarm* (HFSA) seleciona *features* mais importantes e otimiza os hiperparâmetros gama e C de um classificador SVM com núcleo RBF.
FAROOQ et al. (2017)	DA, CCL, CCL em estágio avançado e N.	Segmentação de matéria cinza e *Deep Learning* (RNCs 2D de arquiteturas conhecidas).
REZAEI, YANG e MEINEL (2017)	DA, dois tipos de tumores cerebrais, esclerose múltipla e N.	Extração de fatias axiais, coronais e sagitais de volumes de interesse e *Deep Learning* (rede neural convolucional com arquitetura própria proposta).
SUK e SHEN (2013)	DA, CCL-c, CCL-nc e N.	Representação de *features* baseada em *Deep Learning* (*stacked auto-encoder*), seleção de *features* e SVM com múltiplos núcleos, um para cada modalidade de imagem.
LIU et al. (2015)	DA, CCL-c, CCL-nc[5] e N.	*Deep Learning* (*stacked auto-encoders* com fusão de dados multimodal seguida de uma camada *softmax* de Regressão Logística - RL)
LI et al. (2015)	DA, CCL-c, CCL-nc e N.	Extração de *features* multimodais, análise de componentes principais, *Deep Learning* com *dropout* e SVM com margem suave e núcleo linear.
BHATKOTI e PAUL (2016)	DA e N.	*Deep Learning framework* com fusão de dados multimodal e um *auto-encoder* k esparso modificado.
GARCÍA et al. (2012)	CCL e N.	Método de sobreamostragem proposto para balanceamento dos dados e classificação usando rede neural de contra-propagação.
SEIXAS et al. (2014)	Demência, DA, CCL e N.	RBs com estrutura fornecida e parâmetros aprendidos.
WEAKLEY et al. (2015)	Demência, CCL e N.	Aplicou e avaliou várias combinações de *features* e algoritmos de classificação (NB, árvore de decisão e RL) para dois conjuntos de dados.
MOREIRA e NAMEN (2016)	DA e N.	Avaliou os classificadores NB, RB (com estrutura e parâmetros aprendidos a partir dos dados) e árvore de decisão C4.5 para um mesmo conjunto de dados.
GUERRERO et al. (2016)	CCL e N.	RBs com estrutura fornecida por especialistas e parâmetros aprendidos.
PEREIRA et al. (2017)	Prediz se um paciente com CCL converterá para Demência usando	Para cada janela de tempo: seleção de *features* baseada em correlação, imputação de dados nas observações parciais, sobreamostragem,

[5] Neste trabalho, a janela para a avaliação da conversão foi mudada para 36 meses.

	várias janelas de tempo (2, 3, 4 ou 5 anos).	sintonização de hiperparâmetros para maximizar a área sob a curva ROC e avaliação dos classificadores NB, árvore de decisão C4.5, Floresta Aleatória (RF), SVM com margem suave e núcleos polinomial e RBF, além de k-NN e RL.
BATTISTA, SALVATORE e CASTIGLIONI (2017)	Demência, CCL e N.	Redução de *features* computacionalmente usando Razão de Discriminante de Fisher ou guiada por especialistas, determinação dos melhores preditores e avaliação da classificação usando SVM com diversos núcleos e hiperparâmetros padrão.
CARVALHO et al. (2017a)	Demência, DA, CCL e N.	RBs (com estrutura fornecida e parâmetros aprendidos) integradas a um CDSS.
PEKKALA et al. (2017)	Prediz o risco de um paciente normal desenvolver DA em até 10 anos.	Método de AMS *Disease State Index* (DSI) capaz de indicar os fatores de risco mais importantes.
TRAMBAIOLLI et al. (2011)	DA e N.	Épocas de EEGs são extraídas para diversos canais e a Transformada Rápida de Fourier é aplicada. Posteriormente, são calculadas as coerências entre pares de canais para cada época e tais coerências juntamente com os picos espectrais alimentam um classificador SVM de margem suave e núcleo RBF.
MORABITO et al. (2016)	DA, CCL e N.	*Deep Learning* (RNCs alimentadas por *features* no tempo e na frequência extraídas de EEGs de múltiplos canais).
MIRHEIDARI et al. (2017)	Transtornos neurodegenerativos (incluindo Demência) e transtornos de memória funcional.	Transcrições de conversações entre neurologistas e pacientes com queixas de memória são produzidas manualmente e *features* são automaticamente extraídas destas transcrições incluindo aspectos acústicos, léxicos e semânticos. As *features* são usadas para treinar e avaliar vários algoritmos de classificação incluindo *Perceptron* e RF.
COSTA et al. (2016)	DA e N.	Extração de *features* posturais ajustadas por peso, altura, idade e escolaridade que combinadas com um TNP alimentam alguns algoritmos de classificação que foram avaliados: SVM, *Multilayer Perceptron*, redes neurais com função de ativação RBF e *Deep Belief Networks*.
TRIPOLITI, FOTIADIS e ARGYROPOULOU (2011)	DA e N.	Extração de *features* a partir de RM funcional e outros dados clínicos, seleção de *features* e classificação por RF em versões modificadas.
SUN, LV e TANG (2007)	Demência, CCL e N.	RBs com estrutura e parâmetros aprendidos a partir de *features* extraídas de RM e de dados demográficos e resultados de TNPs.
SEGOVIA et al. (2014)	CCL-c e CCL-nc[6].	*Features* são extraídas de TC e juntamente com resultados de TNPs alimentam um classificador SVM seguindo três paradigmas distintos de integração dos dados. Técnicas distintas de

[6] Neste trabalho, a janela para a avaliação da conversão foi mudada para 36 meses.

		redução de dimensionalidade nas imagens também são aplicadas antes da integração para avaliação juntamente com os paradigmas de integração alternativos.
ZHOU et al. (2014)	DA, CCL amnéstico e não amnéstico[7] e N.	Volumes de estruturas cerebrais de interesse são extraídos de RM e combinados com resultados de TNPs. Posteriormente, *features* são selecionadas usando análise de erro incremental para alimentar diversos classificadores binários (DA versus N, CCL amnéstico versus N e CCL não amnéstico versus N). Tais classificadores são SVM com margem suave e núcleo RBF.
MATTILA et al. (2012)	DA e N.	CDSS que utiliza o índice DSI para predizer a probabilidade do paciente possuir DA com base em dados de RM combinados com resultados de TLs e TNPs.

Tabela 2: Trabalhos relacionados abordando modelos de decisão dinâmicos baseados em *streams* para o auxílio computacional ao diagnóstico de diversas doenças.

Referência	**Doenças**	**Dados**	***Features***	**Algoritmos**	**Desempenho**
ZHANG et al. (2012)	Crônicas monitoradas continuamente	Base não descrita	ECGs, EEGs, temperatura corporal e pressão sanguínea.	VFDT com ponteiros nas folhas	Não descrito
FONG et al. (2013)	Diabetes: 7 níveis BGL	Séries temporais coletadas de 70 pacientes reais durante a terapia de diabetes	Injeções de insulina, refeições, exercícios físicos e medidas de níveis de glicose	VFDT, iOVFDT, Bayes e Perceptron	1) Melhores em termos de Acc: a) VFDT: Acc de 87.43% com σ de 0.040; b) iOVFDT: 86.41% com σ de 0.034; 2) Melhor em termos de AUC e F1: a) Bayes: AUC de 0.847 com σ de 0.05 e F1 de 0.945.
SERHANI, BENHARREF e NUJUM (2014)	11 doenças cardíacas	Coletados por sensores e ECGs de 452 pacientes, base de arritmia do	Temperatura, pressão sanguínea, batimentos cardíacos e	Redes neurais, árvore de decisão, AutoMLP,	RuleModel foi o melhor e conseguiu detectar 11 de 12 classes, incluindo

[7] No CCL não amnéstico, as funções relacionadas à memória estão preservadas e no CCL amnéstico a memória está afetada.

		repositório UCI [8]	ECGs coletados em tempo real por sensores	W-OneR, NB, RuleModel e J48graft	a classe de normalidade.

Tabela 3: Trabalhos relacionados abordando modelos de decisão dinâmicos baseados em retreinamento para o auxílio computacional ao diagnóstico de diversas doenças.

Referência	**Doenças**	**Dados**	***Features***	**Algoritmos**	**Desempenho**
AHMAD, KHAN e MAJEED (2014)	Epilepsia	8736 épocas de EEGs de 54 pacientes	Extraídas de EEGs	DWT e PCA para extração das *features* e SVM com núcleo linear para classificação	Acc de 94.8%, especificidade de 95.7% e sensibilidade de 91.7%. Após retreinamento com correção de 269 épocas, Acc aumentou para 95.12 %.
CHEN et al. (2016)	Epilepsia	EEGs de 944 pacientes com epilepsia e 1280 pacientes normais.	Extraídas de EEGs	WT para extração de *features* e SVM com núcleo RBF para classificação	Considerando percentual de treinamento de 80%, Acc: 97.43% usando CV e 97.63 % para dados de teste em separado.
ROSADO et al. (2015)	Câncer de pele	80 imagens que foram anotadas manualmente por especialistas	Pontuações de critérios de assimetria, borda e cor extraídas de fotos de lesões de pele	Segmentação da lesão, extração de *features* e um algoritmo de classificação não especificado.	Não descrito

Tabela 4: Trabalhos relacionados abordando modelos de decisão dinâmicos para o auxílio computacional ao diagnóstico de DA e transtornos relacionados.

Referência	**Doenças**	**Dados**	***Features***	**Algoritmos**	**Desempenho**
CHAN et al. (1993)	DA	SPECT cerebral de 96 pacientes com DA e 30	Medidas CBF de 16 ROIs de	Rede neural artificial incremental	Acc de 85%

[8] Conjunto de dados de arritmia: http://archive.ics.uci.edu/ml/datasets/Arrhythmia

		pacientes Normais (N)	SPECT		
HUANG e CHEN (2015)	DA e CCL	fMRI cerebral de 350 pacientes: 110 DA, 120 CCL e 120 N	Medidas CBF de ROIs	Clusterização espectral incremental	Após todos os lotes: Acc média de 88.5 %
CHO et al. (2012)	DA, CCL-nc e CCL-c	RM cerebral de 491 pacientes: 160 N, 131 CCL-nc, 72 CCL-c e 128 DA	Dados de espessura cortical	iPCA combinado com iLDA	Após todos os lotes de treinamento: 1) N vs. DA: Acc de 86 %, sensibilidade de 82 % e especificidade de 93%; 2) N vs. CCL-c: acurácia de 82 %, sensibilidade de 66 % e especificidade de 89%; 3) CCL-nc vs. CCL-c: Acc de 70 %, sensibilidade de 63 % e especificidade de 76%.
LEE et al. (2014)	DA	RM cerebral de 117 pacientes: 33 DA e 84 N	Dados de espessura cortical e formato do hipocampo	iPCA combinado com iLDA	Após todos os lotes de treinamento: acurácia de 87.5%, sensibilidade de 97 % e especificidade de 63%.

Tabela 5: Outros trabalhos relacionados abordando modelos de decisão dinâmicos para o auxílio computacional ao diagnóstico de doenças.

Referência	**Doenças**	**Dados**	***Features***	**Algoritmos**	**Desempenho**
SONG et al. (2016)	Câncer de mama	Não identificados de 4640 indivíduos que fizeram triagem em um centro médico universitário	Informação demográfica, sobre a densidade da mama sendo avaliada, sobre a mama oposta e sobre a modalidade utilizada nos exames de imagem	DRA com limiar de decisão sintonizável	TFP: 64.2%, TFN: 1% e Acc: 42%.
VELERT (2012)	Tumores cerebrais: AGG, LGG e MEN	1H MRS de 760 pacientes: 453 com AGG; 162 com LGG e 145 com MEN	Extraídas dos dados de 1H MRS	iGDA e iDBLR	Após todos os lotes de treinamento: Acc de 82 % para o iGDA e Acc de 84.5 % para o iDBLR

CAPÍTULO 3 – FUNDAMENTAÇÃO TEÓRICA

Este capítulo fornece conceitos básicos sobre a tarefa de classificação em geral e particularmente aborda a teoria subjacente aos classificadores que podem ser selecionados pelo processo de aprendizado utilizado para construir o modelo dinâmico proposto. Este processo avalia diversos cenários de pré-processamento e algoritmos de classificação e seleciona, para a construção do modelo, o cenário de pré-processamento e algoritmo de classificação que resulta no melhor desempenho na classificação, considerando diversas medidas.

Os algoritmos de classificação que podem ser selecionados pelo processo de aprendizado incluem técnicas diversificadas, pelo menos um representante de cada um dos seguintes paradigmas: classificadores probabilísticos, baseados em árvore, baseados em similaridade e em funções matemáticas. Além disso, estes classificadores que podem ser selecionados pelo processo de aprendizado são capazes de lidar com observações parciais, ou seja, alguns atributos com valores informados, mas também atributos com valores não informados. Valores não informados para alguns atributos, tanto na base de treinamento quanto na tupla a ser predita, são muito comuns em aplicações médicas e, particularmente, para Demência, DA e CCL, os médicos usualmente aplicam, para cada paciente sendo avaliado, algum subconjunto dos exames mais comuns do centro de saúde em que trabalham.

Os cenários de pré-processamento avaliados pelo processo de aprendizado incluem a aplicação ou não da discretização dos atributos contínuos. Para os cenários com discretização dos atributos contínuos, são avaliados algoritmos de classificação capazes de lidar apenas com atributos discretos, além de algoritmos de classificação capazes de lidar com atributos discretos e contínuos. Por outro lado, para os cenários sem discretização dos atributos contínuos, são avaliados algoritmos de classificação capazes de lidar com atributos discretos e contínuos.

A RB discreta com estrutura fornecida que automatiza a modelagem e o aprendizado de parâmetros propostos por SEIXAS et al. (2014), denotada como BN, e o algoritmo de ponderação de estimadores de uma dependência (denominado A1DE) são algoritmos de classificação capazes de lidar apenas com atributos discretos. Os demais classificadores avaliados pelo processo de aprendizagem são capazes de lidar com ambos os tipos de atributos: contínuos e discretos.

O classificador BN e o modelo híbrido combinando Regressão Logística e Naïve Bayes (denotado como RL-NB ou LR-NB) foram implementados dentro da ferramenta

Waikato Environment for Knowledge Analysis (WEKA)[9]. A implementação dos demais classificadores já estava disponível na ferramenta WEKA.

A Seção 3.1 fornece conceitos básicos sobre a tarefa de classificação em geral. As Seções 3.2 a 3.10 abordam os classificadores que podem ser selecionados pelo processo de aprendizado utilizado para construir o modelo dinâmico proposto. Por fim, a Seção 3.11 descreve métodos de avaliação de classificadores e medidas de desempenho comumente utilizados na literatura.

3.1 O PROBLEMA DA CLASSIFICAÇÃO

Uma tarefa de classificação consiste em um processo de aprendizado de máquina supervisionado em que dados históricos (um conjunto de dados de treinamento contendo elementos previamente atribuídos a um rótulo de classe, ou seja, a um valor que a classe pode assumir) são utilizados como entrada para a construção de um modelo (classificador) que prediz/infere a classe de um novo elemento com classe não conhecida baseando-se nos valores informados para os atributos do classificador.

Considere um vetor de entrada **x** (por exemplo, informação de certo paciente que é usada para tomar uma decisão diagnóstica sobre certa doença ou transtorno) e uma função alvo desconhecida $f: \chi \rightarrow \gamma$ (função ideal para a decisão diagnóstica), onde χ denota o espaço de entrada (conjunto de todos os valores possíveis de **x)** e γ denota o espaço de saída (conjunto de todos os rótulos possíveis, por exemplo, um diagnóstico positivo ou negativo). Considere a existência de um conjunto de dados D com N exemplos de entrada-saída supostamente corretos $(\mathbf{x_1},y_1)$, $(\mathbf{x_2},y_2)$, ..., $(\mathbf{x_N},y_N)$, onde $y_n=f(\mathbf{x_n})$ para n=1,2,...,N (entradas correspondentes a pacientes prévios e a decisão diagnóstica correta para eles em retrospectiva). Os exemplos são frequentemente chamados de exemplos de treinamento, amostras, instâncias ou tuplas de treinamento. Um algoritmo de aprendizado usa o conjunto D para derivar uma função $g: \chi \rightarrow \gamma$ que aproxima f. O algoritmo escolhe g a partir de um conjunto de funções candidatas consideradas, chamado de conjunto de hipóteses H. Tal conjunto pode ser visto como o conjunto de todas as funções que o algoritmo de aprendizado pode escolher para melhor representar os dados de treinamento.

[9] A WEKA foi inicialmente desenvolvida pela universidade de Waikato, na Nova Zelândia, mas, atualmente vários desenvolvedores contribuem para seu código-fonte aberto que é passível de estudo e alteração. Ela agrega algoritmos de diferentes paradigmas utilizados na mineração de dados e no aprendizado de máquina. Para maiores informações, acessar https://www.cs.waikato.ac.nz/ml/weka/.

Quando um novo paciente é avaliado, o médico pode basear sua decisão diagnóstica em g (a hipótese que o algoritmo de aprendizado escolheu). Note que a decisão diagnóstica não se baseia em f, a função alvo ideal, pois esta é desconhecida. A decisão será tão boa quanto g conseguir aproximar f. Para isto, o algoritmo escolhe g, que melhor aproxima f, com base nos exemplos de treinamento de pacientes diagnosticados anteriormente, com a suposição de que tal aproximação continue a valer para novos pacientes. O algoritmo de aprendizado explora o conjunto inteiro de hipóteses H, procurando por algum h $\in$ H que minimize alguma medida de erro, por exemplo, a taxa de erro dentro da amostra (fração de D onde f e h discordam):

$$E_{in}(h) = \frac{1}{N}\sum_{n=1}^{N}[\![h(\mathbf{x_n}) \neq f(\mathbf{x_n})]\!] \quad (1)$$

Onde $[\![\text{expressão}]\!]$ resulta em 1 se a expressão é verdadeira ou 0 caso contrário.

Nas aplicações práticas, os dados a partir dos quais se aprende não são gerados por uma função alvo f determinística. Ao contrário, eles contêm discrepâncias de tal maneira que a saída não é unicamente determinada pela entrada. Por exemplo, no conjunto de dados de treinamento, dois pacientes com atributos idênticos podem ter diagnósticos distintos. Considerando tal situação, a saída Y é modelada como uma variável aleatória que é afetada (e não determinada) pela entrada. Formalmente, tem-se uma distribuição alvo $P(Y|\mathbf{X})$ ao invés de $y=f(\mathbf{x})$, em que variáveis aleatórias são denotadas por letra maiúscula, valores que tais variáveis podem assumir são denotados por letras minúsculas e vetores são representados em negrito. Desta forma, considera-se que cada elemento do conjunto de dados é gerado por uma distribuição conjunta $P(\mathbf{X}, Y) = P(\mathbf{X})P(Y|\mathbf{X})$. A Figura 1 ilustra o problema da classificação de acordo com as convenções adotadas.

O paradigma Bayesiano (BERTSEKAS e TSITSIKLIS, 2002), que pode ser adotado pelo algoritmo de aprendizado, constrói um modelo em que uma hipótese para f maximiza a distribuição a posteriori P(h=f|D):

$$P(h = f|D) \sim P(D|h = f)\, P(h = f) \quad (2)$$

Onde P(D|h=f) é a função de verossimilhança estimada a partir dos dados de treinamento e P(h=f) é a distribuição a priori sobre o conjunto de hipóteses refletindo as probabilidades das diferentes hipóteses representarem a função alvo correta. A distribuição a priori representa as crenças sobre o conjunto de hipóteses antes da observação de qualquer dado e tais crenças são atualizadas considerando os dados de treinamento através da maximização da distribuição a posteriori.

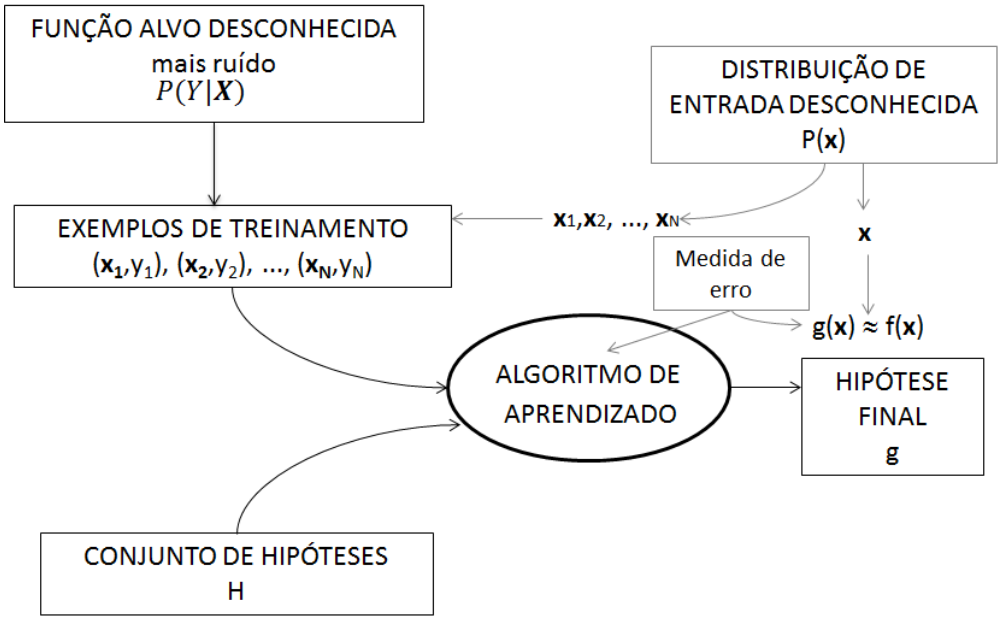


Figura 1: O problema da classificação. Adaptado de: ABU-MOSTAFA, MAGDON-ISMAIL e LIN (2012).

3.2 REDES BAYESIANAS

Redes Bayesianas (RBs) (PEARL, 1988; KOLLER, FRIEDMAN e BACH, 2009) são modelos gráficos probabilísticos capazes de lidar com incerteza e realizar inferência baseando-se em observações parciais, ou seja, contendo evidências (valores informados para certos atributos), mas também atributos com valores não informados. Além disso, as RBs são modelos interpretáveis capazes de melhorar a comunicação com especialistas do domínio de interesse.

Uma RB é totalmente especificada por uma parte qualitativa e uma parte quantitativa. A parte qualitativa de uma RB é um grafo acíclico direcionado (DAG - *Directed Acyclic Graph*) que compreende:

1. Nós representando variáveis aleatórias.
2. Um conjunto de ligações direcionadas ou setas que conectam pares de nós. O significado de uma seta partindo de um nó V_1 para um nó V_2, é que a variável V_1 tem uma influência direta sobre a variável V_2 e, neste caso, V_1 é chamado pai de V_2.

Cada nó em uma RB possui uma distribuição condicional, isto é, uma distribuição de probabilidades condicionais (CPD - *Conditional Probabilities Distribution*), que quantifica os efeitos que os pais possuem sobre o nó. A parte quantitativa de uma RB é o conjunto de todas as CPDs. A parte qualitativa de uma RB também é comumente chamada de estrutura da RB e

a parte quantitativa de parâmetros da RB. Nas RBs discretas, somente variáveis aleatórias discretas são consideradas e, se existem variáveis contínuas em um domínio, essas são discretizadas antes do processo de aprendizado.

PRADHAN et al. (1994) propuseram um *template* de estrutura em três níveis: o primeiro nível contendo nós representando fatores predisponentes, o segundo nível contendo o nó representando a doença ou transtorno a diagnosticar e o terceiro nível contendo nós representando sintomas, sinais e resultados de testes aplicados sobre os pacientes. A Figura 2 ilustra a estrutura de uma RB exemplo seguindo tal *template* para diagnóstico de Demência. As CPDs não estão ilustradas na Figura 2, os nós do primeiro nível representam fatores predisponentes, o nó Y representa a classe e os nós do terceiro nível representam TNPs que podem ser utilizados no diagnóstico de Demência, os quais serão detalhados no Capítulo 6.

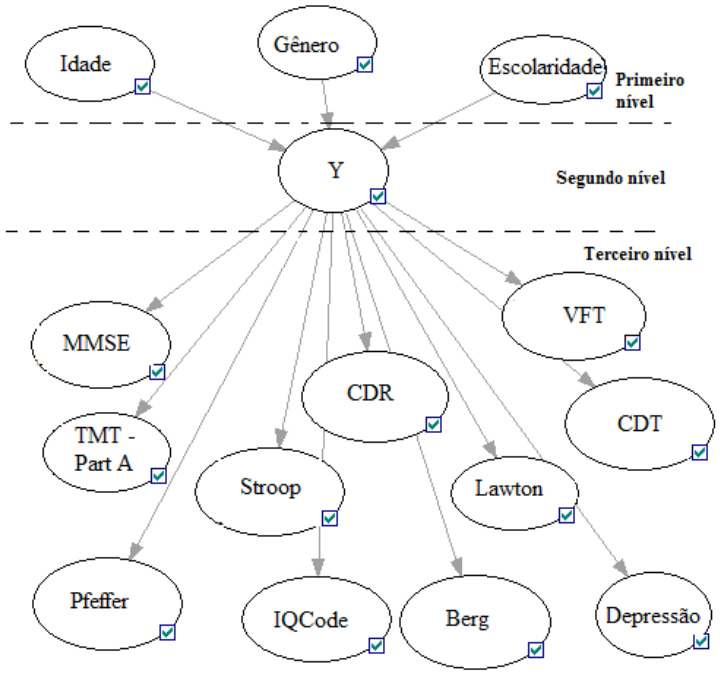


Figura 2: Estrutura de uma RB exemplar construída para diagnóstico de Demência.

Considere que uma RB contenha um conjunto de nós ou variáveis aleatórias $V^G = \langle V_1, V_2, \dots, V_d, V_{d+1} \rangle$ em que uma destas variáveis representa a classe Y e as demais variáveis representam os d atributos de **X.** Suponha que V_i seja a i-ésima variável aleatória contendo r_i estados discretos. Suponha também que v_i^k é o k-ésimo estado de V_i. Tal RB consiste em um par $\langle G,\Theta \rangle$, onde G representa o DAG e Θ o conjunto de probabilidades condicionais $\Theta = \langle \theta_{ijk} : \forall ijk \rangle$, onde $\theta_{ijk} = Pr(V_i = v_i^k | pais(V_i) = p_i^j)$, onde p_i^j representa a j-ésima combinação de estados dos nós pais de V_i.

Dado um conjunto de dados de treinamento composto por amostras independentes e identicamente distribuídas (i.i.d.), os parâmetros da RB que melhor representam tais dados para uma estimativa de máxima verossimilhança *Maximum Likelihood Estimation* (MLE) θ_{ijk}^{MLE} são comumente utilizados como parâmetros da RB. Quando os nós de uma RB discreta podem assumir dois ou mais estados, a função de verossimilhança segue uma distribuição multinomial, a distribuição a priori segue uma distribuição de Dirichlet com parâmetros α_{ijk} atribuídos inicialmente de acordo com as crenças a priori sobre as CPDs e a distribuição a posteriori também segue uma distribuição de Dirichlet com parâmetros atualizados após a observação dos exemplos. Um método de aprendizado comumente aplicado no caso de observações parciais na base de treinamento é o algoritmo iterativo *Expectation-Maximization* (EM) (DEMPSTER et al., 1977; LAURITZEN e SPIEGELHALTER, 1988). Tal algoritmo contém dois passos: E (*Expectation*) e M (*Maximization*). No passo E, valores esperados para os valores não informados são estimados para completar os dados usando os valores informados e a estimativa atual para θ; depois disto, dados os valores estimados pelo passo E, o passo M obtém o θ que maximiza a verossimilhança e o atribui ao θ que será usado pelo passo E da próxima iteração. O algoritmo continua até alcançar a convergência atingindo um máximo local ou outro critério de parada (por exemplo, um número máximo de iterações). Como resultado, os parâmetros da RB são obtidos (aprendidos) a partir dos dados. As entradas para o algoritmo EM são: a base de treinamento e os valores iniciais para α_{ijk}. Assumindo inicialmente que os parâmetros da rede seguem uma distribuição uniforme ou que não se possui informação a priori sobre eles, α_{ijk} são igualados a 1 (COOPER e HERSKOVITS, 1992) de forma que o EM começa com todos os parâmetros da rede tomados da distribuição uniforme. A saída do EM são os parâmetros da RB aprendidos a partir dos dados de treinamento.

A inferência Bayesiana (utilizada na classificação/predição realizada por RBs) pode ser vista como um processo de atualização das distribuições marginais de alguma variável de interesse V_i dada a ocorrência de valores de outras variáveis que são conhecidas (observadas), as chamadas evidências. Considere a RB da Figura 2 usada para o diagnóstico de Demência após o processo de aprendizado dos parâmetros. O nó de diagnóstico representando a classe, o nó Y, pode assumir dois estados com o valor para a classe denotado por $y \in \{0,1\}$, em que 0 representa o diagnóstico negativo para Demência e 1 representa o diagnóstico positivo para Demência. Para predizer se um paciente possui ou não Demência, algumas evidências devem ser atribuídas a nós da RB e a inferência Bayesiana deve ser realizada para resolver

P(Y=1|evidências). Por exemplo, se para o paciente sendo avaliado, foi coletada informação quanto à idade, gênero, escolaridade e a pontuação obtida para o TNP MMSE, as evidências correspondentes para os nós Idade, Gênero, Escolaridade e MMSE são atribuídas. Depois disto, a inferência Bayesiana computa a probabilidade do paciente possuir diagnóstico positivo para Demência.

Formalmente, para realizar a inferência na RB, deseja-se computar a distribuição condicional de $V_i \in \mathbf{V} \setminus \mathbf{E}$, dada uma observação (evidência) $\mathbf{E} = \mathbf{e}$. Tal distribuição é denotada por $p(v_i|\mathbf{e})$ sendo expressa por:

$$p(v_i|\mathbf{e}) = \frac{p(v_i,\mathbf{e})}{p(\mathbf{e})} \qquad (3)$$

Onde as letras em negrito **V, E, e** representam vetores; o símbolo \ denota a exclusão do subconjunto da esquerda do conjunto da direita; $\mathbf{V}= [V_1, \ldots , V_{d+1}]$ denota o conjunto de todas as variáveis ou nós da RB; $\mathbf{E} \subset \mathbf{V}$ denota o subconjunto de variáveis observadas em **V**; e **e** denota os valores observados em **E.**

Em (3), uma vez que o denominador não depende de v_i, a tarefa de inferência consiste em obter $p(v_i,\mathbf{e})$ e depois normalizar. Um algoritmo de inferência "força bruta" pode ser expresso como (LANGSETH et al., 2009):

(1) Obter a distribuição conjunta $p(v_1, ..., v_{d+1})$:

$$p(v_1, \cdots, v_{d+1}) = \prod_{i=1}^{d+1} p(v_i|\text{pais}(v_i)) \qquad (4)$$

(2) Restringir $p(v_1, ..., v_{d+1})$ aos valores **e**, obtendo $p(v_1, ..., v_{d+1}, \mathbf{e})$

(3) Computar $p(v_i, \mathbf{e})$ a partir de $p(v_1, ..., v_{d+1}, \mathbf{e})$ marginalizando toda variável excetuando-se V_i.

Um problema com este método é que a distribuição conjunta é geralmente grande e difícil de gerenciar. LANGSETH et al. (2009) ilustram o uso do algoritmo de eliminação de variáveis, que através da organização das operações entre as distribuições condicionais da RB, permite a eliminação da manipulação de distribuições desnecessariamente grandes. Ainda, ressaltam que o processo de inferência exato requer o uso de três operações básicas: restrição, combinação e eliminação. Fundamentalmente, deve-se assegurar que tais operações possam ser realizadas analiticamente e, do ponto de vista prático, é desejável que uma estrutura de dados única represente todos os resultados intermediários de tais operações. Não é difícil encontrar exemplos em que tais requisitos não sejam cumpridos, particularmente quando

algumas variáveis do domínio são contínuas. Uma solução frequentemente empregada consiste na discretização das variáveis contínuas.

KORB e NICHOLSON (2011) apresentam um algoritmo para inferência exata e exemplificam sua aplicação em RBs com variáveis discretas para o caso particular de RBs do tipo *polytree*, as quais possuem um único caminho para conectar dois nós quaisquer da rede. Além disso, eles descrevem e ilustram um algoritmo baseado em fluxo de mensagens entre os nós da rede (*belief propagation*), o qual promove a atualização das probabilidades de um nó através do recebimento de mensagens vindas de seus pais e/ou filhos (as quais contêm evidências) e, após a atualização das probabilidades do nó, há o envio de novas mensagens para seus pais e/ou filhos até que não haja mais mensagens para propagar e nós para atualizar. Além disso, descrevem o processo de inferência exata para RBs multiconectadas através de métodos de clusterização que as fazem recair no caso das *polytrees*, ilustrando-o para RBs discretas.

Um dos classificadores que pode ser escolhido pela metodologia proposta pela tese é uma RB discreta com estrutura fornecida, composta por três níveis como proposto por PRADHAN et al. (1994) e com aprendizado de parâmetros através de EM já que a base de treinamento possui observações parciais. Tal método automatiza a modelagem proposta por SEIXAS et al. (2014). Tanto o aprendizado de parâmetros quanto a inferência Bayesiana são realizadas invocando-se a API GeNIe/SMILE[10]. No aprendizado, assume-se incialmente que os parâmetros da rede seguem uma distribuição uniforme sendo a convergência do EM alcançada quando a estimativa de máxima verossimilhança alcança um máximo local. Para a inferência Bayesiana, tal API normalmente utiliza inferência exata aplicando clusterização.

3.3 NAÏVE BAYES

Naïve Bayes (NB) (JOHN e LANGLEY, 1995) é um classificador Bayesiano muito popular que assume que, dado o valor da classe, os atributos são condicionalmente independentes entre si. NB é capaz de lidar com atributos contínuos e discretos e com observações parciais. A Figura 3 ilustra a estrutura de um modelo NB.

Suponha que se deseja estimar, a partir de um conjunto de dados de treinamento D contendo N tuplas com d atributos, a probabilidade $P(y|\mathbf{x})$ de um exemplo de teste $\mathbf{x} = [a_1, \ldots, a_d]$ pertencer à classe y, onde a_i é o valor do i-ésimo atributo e $y \in \{c_1,\ldots,c_K\}$, em que c_k representa os valores possíveis para a classe.

Da definição de probabilidade condicional, tem-se:

[10] https://download.bayesfusion.com/_les.html?category=Academia/

$$P(y|\mathbf{x}) = P(y,\mathbf{x})/P(\mathbf{x}) \qquad (5)$$

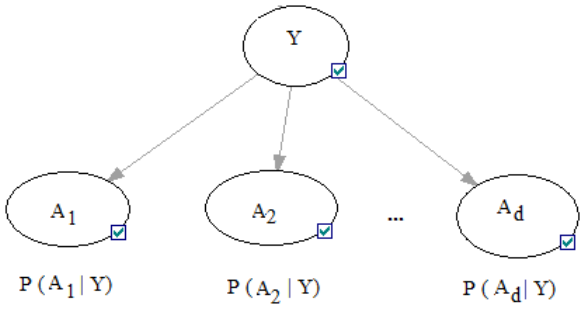


Figura 3: Modelo Naïve Bayes.

Como o denominador em (5) é constante para todas as classes, é suficiente determinar $P(y,\mathbf{x})$, pois a classe será escolhida de acordo com o maior valor de $P(y|\mathbf{x})$. Aplicando-se o teorema de *Bayes*, $P(y,\mathbf{x})$ pode ser expresso como:

$$P(y,\mathbf{x}) = P(y)P(\mathbf{x}|y) \qquad (6)$$

Como todos os atributos são independentes entre si dado o valor da classe, tem-se:

$$P(\mathbf{x}|y) = \prod_{i=1}^{d} P(a_i|y) \qquad (7)$$

Substituindo-se (7) em (6), obtém-se a expressão que o modelo NB utiliza para classificar o exemplo de teste:

$$\hat{P}_{NB}(y,\mathbf{x}) = \hat{P}(y)\ \prod_{i=1}^{d} \hat{P}(a_i|y) \qquad (8)$$

Onde: $\hat{P}(e)$ denota a estimativa de probabilidade de ocorrência de um evento e. Em (8), $\hat{P}(y)$ e $\prod_{i=1}^{d} \hat{P}(a_i|y)$ são derivados diretamente de D usando contagem e proporções.

Se A_i é um atributo contínuo, considera-se que este possui uma distribuição Gaussiana com μ e σ derivados dos valores de A_i nas tuplas de treinamento classificadas como y e $\hat{P}(a_i|y)$ é estimada a partir da Função de Densidade de Probabilidade (FDP) correspondente.

3.4 PONDERAÇÃO DE ESTIMADORES DE UMA DEPENDÊNCIA

WEBB et al. (2005) propuseram *Averaged One-Dependence Estimators* (AODE), também conhecido como A1DE, formado por um conjunto de classificadores com uma dependência que são aprendidos a partir dos dados. A predição é produzida agregando-se as predições de todos os classificadores qualificados. Um estimador de L dependência significa que a probabilidade de cada atributo é condicionada à variável de classe e até L outros

atributos. Sendo assim, A1DE relaxa as pressuposições de independência do modelo NB.

Suponha que se queira estimar, a partir de uma base de treinamento D contendo N tuplas com d atributos, a probabilidade $P(y|\mathbf{x})$ de um exemplo de teste $\mathbf{x} = [a_1, \ldots, a_d]$ pertencer à classe y, onde a_i é o valor do i-ésimo atributo e $y \in \{c_1,\ldots,c_K\}$, em que c_k representa os valores possíveis para a classe.

A1DE usa o conceito de "super pai" considerando cada atributo independente dos demais atributos dados os valores da classe e de outro atributo, o "super pai", a_α, o que representa uma independência condicional mais fraca em relação ao NB. A1DE usa:

$$P(y, \mathbf{x}) = P(y, a_\alpha).P(\mathbf{x}|y, a_\alpha) \qquad (9)$$

Junto com a pressuposição de independência condicional:

$$P(\mathbf{x}|y, a_\alpha) = \prod_{i=1}^{d} P(a_i|y, a_\alpha) \qquad (10)$$

A1DE busca fazer uma média usando todas as estimativas de $P(y|\mathbf{x})$ produzidas usando "super pais" distintos:

$$\hat{P}(y, \mathbf{x}) = \sum_{\alpha=1}^{d} \hat{P}(y, a_\alpha)\, \hat{P}(\mathbf{x}|y, a_\alpha)/d \qquad (11)$$

Contudo, de fato, A1DE somente usa as estimativas de probabilidades para as quais exemplos relevantes ocorrem nos dados:

$$\hat{P}_{A1DE}(y, \mathbf{x}) = \begin{cases} \frac{\sum_{\alpha=1}^{d} \delta(a_\alpha)\hat{P}(y,a_\alpha)\hat{P}(\mathbf{x}|y,a_\alpha)}{\sum_{\alpha=1}^{d} \delta(x_\alpha)}, & \sum_{\alpha=1}^{d} \delta(a_\alpha) > 0 \\ \hat{P}_{NB}(y, \mathbf{x}), & \text{caso contrário} \end{cases} \qquad (12)$$

Em (12), se o valor-atributo a_α está presente nos dados, $\delta(a_\alpha)$ é 1, caso contrário é 0. Portanto, A1DE faz a média sobre todos os "super pais" cujos valores ocorrem nos dados e usa a estimativa NB se tais valores de "super pais" não ocorrem nos dados.

3.5 MODELO HÍBRIDO DE REGRESSÃO LOGÍSTICA – NAÏVE BAYES

CARVALHO et al. (2017b) propuseram, baseando-se no descrito por TAN et al. (2016), um modelo híbrido combinando Regressão Logística e Naïve Bayes (*Hybrid Logistic Regression-Naïve Bayes Model*, LR-NB ou RL-NB) que modela uma RB com estrutura tal como proposta por PRADHAN et al. (1994). Tal modelo híbrido é capaz de conter nós discretos e contínuos e lidar com observações parciais. A Figura 4 ilustra este modelo. A parte da RB incluindo o nó Y (nó classe) e os nós F_1, F_2, … , F_n, formam a parte Regressão Logística (RL) do modelo e a parte da RB incluindo o nó Y e os nós E_1, E_2, … , E_m formam a

parte NB do modelo. Como ilustrado pela Figura 4, os atributos da parte RL do modelo são condicionalmente independentes dos atributos da parte NB do modelo, dada a classe Y. Para aprender parâmetros da distribuição condicional de Y, dados os atributos **F** da parte RL, os atributos **E** da parte NB são irrelevantes nesta tarefa. Portanto, é possível usar métodos de estimativa de parâmetros baseados em RL padrão. Similarmente, para aprender os parâmetros dos atributos **E** da parte NB do modelo híbrido, atributos **F** da parte RL são irrelevantes nesta tarefa e, portanto, é viável usar métodos de estimativa de parâmetros baseados em NB padrão para estimar tais parâmetros.

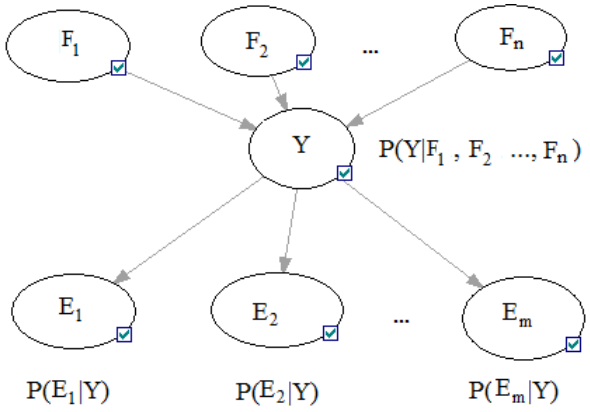


Figura 4: Modelo Híbrido de Regressão Logística-Naïve Bayes. Fonte: CARVALHO et al. (2017b).

Suponha que Y é a variável de classe binária a ser predita assumindo valores y ∈ {0,1} e F_1, F_2, … , F_n, são atributos com valores reais usados para predizer Y. Na prática, estes atributos podem ser contínuos ou booleanos (com valor 0 ou 1). Se existe um atributo categórico com k valores distintos, deve-se representar este atributo no modelo usando k-1 atributos booleanos. Para um modelo RL, tem-se:

$$\text{odds}(Y = 1|f_1, \dots, f_n) = \exp\left(\beta_0 + \sum_{j=1}^{n} \beta_j f_j\right) \quad (13)$$

Onde β_0, β_j são os coeficientes da RL, os quais são estimados pelo aprendizado a partir dos dados e *odds* denota a razão entre a probabilidade de um evento ocorrer e a probabilidade desse mesmo evento não ocorrer.

Como visto anteriormente, NB é um classificador probabilístico baseado no Teorema de *Bayes*. Ele pressupõe que todos os atributos são condicionalmente independentes entre si, dada a variável de classe. Supondo que Y é a variável de classe binária assumindo valores y ∈ {0,1} a ser predita baseando-se em uma observação de um subconjunto de m atributos **E** = (E_1, E_2, ... , E_m), em que tais atributos podem ser contínuos ou discretos, para um modelo NB, tem-se:

$$\text{odds}(Y = 1|e_1, \ldots, e_m) = \text{odds}(Y = 1) \prod_{i=1}^{m} \text{lr}(e_i, Y = 1) \quad (14)$$

Onde $\text{lr}(e_i, Y = 1) = \frac{P(e_i|Y=1)}{P(e_i|Y=0)}$. Portanto, a razão entre as probabilidades a posteriori de Y = 1 é igual à razão entre as probabilidades a priori de Y = 1 multiplicada pelas razões entre probabilidades dos atributos observados para Y = 1. Se um determinado atributo não é observado, sua razão é 1.

Após a eliminação dos atributos **F** da parte RL do modelo híbrido RL-NB, o que sobra é um modelo NB cuja distribuição a posteriori de Y (dado **F**=**f**) é dada por (13). Portanto, a distribuição a posteriori de Y dado que **F** = **f** e **E** = **e** usando (14) será:

$$\text{odds}(Y = 1|e_1, \ldots, e_m, f_1 \ldots, f_n) = \text{odds}(Y = 1|f_1, \ldots, f_n) \prod_{i=1}^{m} \text{lr}(e_i, Y = 1) \quad (15)$$

Isolando o último termo em (14) e substituindo em (15), obtém-se:

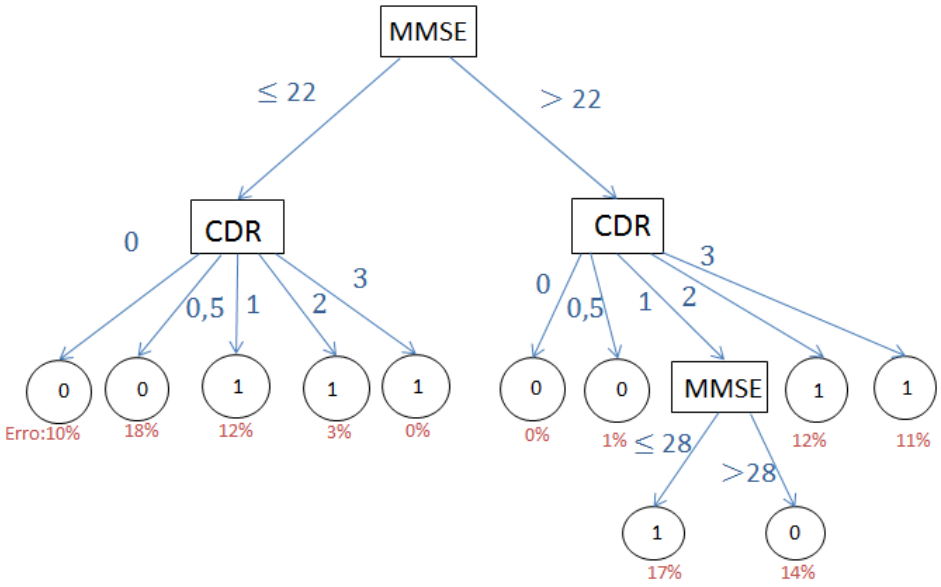
$$\text{odds}(Y = 1|\mathbf{e}, \mathbf{f}) = \frac{1}{\text{odds}(Y=1)} \text{odds}(Y = 1|\mathbf{e})\ \text{odds}(Y = 1|\mathbf{f}) \quad (16)$$

Onde **e** denota o vetor contendo os valores observados para os atributos do conjunto **E** e **f** denota o vetor contendo os valores observados para os atributos do conjunto **F**.

A Equação (16) é utilizada para realizar a inferência no modelo híbrido RL-NB. A parte NB pode naturalmente lidar com valores não informados, enquanto, para os atributos do conjunto **F**, a parte RL substitui atributos não informados pela média (para atributos contínuos) ou pela moda (para atributos discretos) antes do aprendizado dos betas em (13).

3.6 ÁRVORE DE DECISÃO C4.5

Uma árvore de decisão é uma estrutura em que cada nó interno (não folha) denota um teste realizado sobre um atributo, cada ramo representa uma saída para um teste e cada folha (ou nó terminal) possui um rótulo de classe. Cada caminho partindo do nó raiz da árvore até uma folha representa uma regra, definida como um conjunto de condições que implicam o valor de classe encontrado na folha em questão. A árvore é construída de tal forma que, para um mesmo registro (tupla) com todos os valores de atributo informados, existe um e somente um caminho da raiz até a folha. A Figura 5 exemplifica uma árvore de decisão para diagnóstico de Demência e as regras de classificação associadas em que a classe assume o valor $y \in \{0,1\}$, em que 0 representa o diagnóstico negativo para Demência e 1 representa o diagnóstico positivo para Demência.



Regra 1: $(MMSE \leq 22)\text{^}(CDR = 0) \rightarrow negativo$ $(erro\ 10\%)$

Regra 2: $(MMSE \leq 22)\text{^}(CDR = 0{,}5) \rightarrow negativo$ $(erro\ 18\%)$

Regra 3: $(MMSE \leq 22)\text{^}(CDR = 1) \rightarrow positivo$ $(erro\ 12\%)$

Regra 4: $(MMSE \leq 22)\text{^}(CDR = 2) \rightarrow positivo$ $(erro\ 3\%)$

Regra 5: $(MMSE \leq 22)\text{^}(CDR = 3) \rightarrow$ positivo $(erro\ 0\%)$

Regra 6: $(MMSE > 22)\text{^}(CDR = 0) \rightarrow negativo$ $(erro\ 0\%)$

Regra 7: $(MMSE > 22)\text{^}(CDR = 0{,}5) \rightarrow negativo$ $(erro\ 1\ \%)$

Regra 8: $(MMSE > 22)\text{^}(CDR = 1)\text{^}(MMSE \leq 28) \rightarrow positivo$ $(erro\ 17\%)$

Regra 9: $(MMSE > 22)\text{^}(CDR = 1)\text{^}(MMSE > 28) \rightarrow negativo$ $(erro\ 14\%)$

Regra 10: $(MMSE > 22)\text{^}(CDR = 2) \rightarrow positivo$ $(erro\ 12\%)$

Regra 11: $(MMSE > 22)\text{^}(CDR = 3) \rightarrow positivo$ $(erro\ 11\%)$

Figura 5: Exemplo de árvore de decisão para diagnóstico de Demência e as regras associadas.

O aprendizado de árvores de decisão a partir de tuplas de treinamento com classe conhecida (rotuladas) é chamada de indução. Existem diversos algoritmos para indução de árvores de decisão sendo o ID3 (QUINLAN, 1986) e o C4.5 (QUINLAN, 1993) alguns exemplos. Estes algoritmos adotam uma heurística gulosa baseada em dividir para conquistar na qual a árvore é construída de cima para baixo de forma recursiva e o conjunto de treinamento é particionado em conjunto menores na medida em que árvore é construída. O algoritmo básico está mostrado na Figura 6.

Algoritmo: gerar_arvore_decisao. Gera uma árvore de decisão a partir das tuplas de treinamento de uma partição de dados D.
Entrada:

- Partição de dados, D, conjunto de tuplas de treinamento com rótulos de classe associados.
- lista_atributos, o conjunto de atributos candidatos.
- metodo_selecao_atributos, um procedimento para determinar o critério de partição (criterio_corte) que melhor particiona as tuplas de dados de entrada em classes individuais.

Saída: uma árvore de decisão.

Método:

(1) crie um nó NO;

(2) **se** tuplas em D são todas da mesma classe, y, **então**

(3) retorna NO como um nó folha rotulado com a classe y;

(4) **se** lista_atributos está vazia **então**

(5) retorna NO como um nó folha rotulado com a classe majoritária em D;

(6) aplique metodo_selecao_atributo (D, lista_atributos) para encontrar o melhor criterio_corte;

(7) nomeie NO com o criterio_corte;

(8) **se** atributo_corte é discreto **e**

cortes de várias maneiras são permitidos , **então** //não restrito a árvores binárias

(9) lista_atributos ← lista_atributos – atributo_corte; //remove atributo de corte

(10) **para cada** saída j do criterio_corte

// particione as tuplas e cresça as sub-árvores de cada partição

(11) faça D_j ser o conjunto de tuplas de dados em D satisfazendo a saída j; //partição

(12) **se** D_j está vazio **então**

(13) anexe uma folha nomeada coma classe majoritária em D a NO;

(14) **senão** anexe **gerar_arvore_decisao** (D_j, lista_atributos) a NO;

final-para

(15) retorne NO;

Figura 6: Algoritmo básico para indução de árvore de decisão a partir das tuplas de treinamento. Fonte: HAN, PEI e KAMBER (2011).

O algoritmo da Figura 6 é chamado com três parâmetros: D, *lista_atributos* e *metodo_selecao_atributos*. D é uma partição de dados. Inicialmente, D é o conjunto completo de tuplas de treinamento e seus rótulos de classe associados. O parâmetro *lista_atributos* é a lista de atributos descrevendo as tuplas e *metodo_selecao_atributos* especifica uma heurística para selecionar o atributo que melhor discrimina as tuplas de acordo com a classe. Trata-se de um procedimento para determinar um critério de partição (*criterio_corte*) que melhor particiona as tuplas de dados de entrada em classes individuais. Este critério consiste em um atributo de corte (*atributo_corte*) e um ponto de corte (*ponto_corte*) ou um subconjunto de corte (*subconjunto_corte*).

Para a árvore de decisão C4.5, o método *metodo_selecao_atributos* seleciona, de *lista_atributos,* o atributo com maior razão de ganho (*gain ratio*) que é uma forma normalizada do ganho de informação e uma medida baseada em entropia cujo cálculo é detalhado nos próximos parágrafos. O objetivo é minimizar a quantidade de testes necessários para classificar uma tupla. A saída final do algoritmo é a árvore de decisão aprendida a partir das tuplas de treinamento, permitindo que uma nova tupla seja classificada de acordo com seus atributos.

O ganho de informação de um dado atributo representa a redução esperada na entropia causada por particionar os registros de acordo com os valores do respectivo atributo sendo dado por:

$$\text{Ganho\_informação}(A) = E(S) - E(S, A) \qquad (17)$$

Onde:

E(S) é a entropia de uma partição S da base contendo *s* registros que pertencem a *K* classes sendo dada por:

$$E(S) = -\sum_{k=1}^{K} p_k \log_2(p_k) \quad (18)$$

Em (18), p_k é a proporção de registros de S pertencente à k-ésima classe.

Em (17), E(S,A) é a entropia considerando-se o particionamento de S de acordo com os valores do atributo A sendo dada por:

$$E(S, A) = \sum_{v \in \text{Domínio (A)}} \frac{|S_v|}{|S|} E(S_v) \quad (19)$$

Em (19), S_v é a partição de S que contém o valor v em A e |Partição| denota o número de registros da partição.

Por sua vez, a razão de ganho é dada por:

$$\text{Razão\_ganho}(A) = \frac{\text{Ganho\_informação (A)}}{-\sum_{v \in \text{Domínio (A)}} \frac{|S_v|}{|S|} \log_2\left(\frac{|S_v|}{|S|}\right)} \quad (20)$$

O algoritmo básico mostrado na Figura 6 será explicado a seguir. A árvore inicia com um nó denotado como NO representando as tuplas de treinamento em D (passo 1). Se todas as tuplas em D possuírem a mesma classe, então o nó se tornará uma folha rotulada com esta classe (passos 2 e 3). Caso contrário, o algoritmo invoca *metodo_selecao_atributos* para determinar o critério de partição, ou seja, qual atributo será testado em NO ao determinar a melhor maneira de particionar as tuplas em D em classes individuais (passo 6) e qual teste será escolhido determinando quais ramos crescer a partir de NO em relação às saídas do teste. Mais especificamente, o critério de partição indica o atributo de corte e também indica um ponto de corte (se o atributo for contínuo) ou um subconjunto de corte (se o atributo for categórico). O critério de partição é determinado de forma que, idealmente, as partições resultantes de cada ramo sejam as mais puras possíveis, ou seja, possuam a maioria das tuplas pertencentes à mesma classe (entropia reduzida, maior ganho). Então, o NO é nomeado com o critério de partição, o qual serve como um teste no nó (passo 7). Um ramo cresce a partir de NO para cada uma das saídas do critério de partição. As tuplas são particionadas de acordo (passos 10 e 11). O algoritmo usa o mesmo processo recursivamente para formar uma árvore de decisão para as tuplas de cada partição resultante D_j (passo 14). O particionamento recursivo pára quando uma das seguintes condições for verdade:

1. Todas as tuplas da partição pertencerem à mesma classe (passos 2 e 3); ou
2. Não existirem mais atributos restantes para os quais as tuplas possam ser adicionalmente particionadas (passo 4) e, neste caso, o nó é convertido em uma folha com rótulo da classe majoritária da partição (passo 5) e uma estimativa de erro associada é armazenada. Tal estimativa de erro é calculada através da proporção de tuplas de treinamento representadas pela folha que pertencente à classe majoritária (1- probabilidade da folha representar a classe majoritária); ou
3. Não existirem mais tuplas para um dado ramo, isto é, uma partição D_j está vazia (passo 12) e, neste caso, uma folha é criada com rótulo igual ao da classe majoritária em D (passo 13).

A árvore de decisão resultante é retornada no passo 15.

O foco desta seção é a árvore de decisão C4.5. Ao contrário da árvore de decisão ID3, a árvore de decisão C4.5 é capaz de lidar com observações parciais e também com atributos contínuos. Por isso, a árvore de decisão C4.5 é um dos classificadores que pode ser escolhido pela metodologia proposta na presente tese, pois, além de lidar com observações parciais, pode ser avaliada tanto nos cenários de pré-processamento com discretização como nos cenários de pré-processamento sem discretização como será visto no próximo capítulo.

Para atributos contínuos, o método de seleção de atributos do C4.5 determina tanto o atributo quanto o ponto de corte para o teste que gera duas partições: uma partição com tuplas satisfazendo a condição de que o valor do atributo é menor ou igual ao valor do ponto de corte e a outra partição com tuplas satisfazendo a condição de que o valor do atributo é maior que o valor do ponto de corte. Para selecionar o ponto de corte para cada atributo contínuo, o método coloca em ordem crescente todos os valores presentes na base para o atributo em questão e determina o ganho de informação para cada ponto de corte candidato, ou seja, para cada valor presente na base imediatamente inferior ao ponto médio entre valores adjacentes presentes na base. Então, seleciona o ponto de corte que mais reduz a entropia da base, ou seja, o que produz maior ganho. Com o ponto de corte de cada atributo contínuo já determinado, é selecionado o atributo com maior razão de ganho.

Para a indução da árvore de decisão C4.5, no caso de observações parciais, ou seja, tuplas possuindo alguns atributos com valores não informados, o ganho de informação passa a ser calculado como:

$$\text{Ganho\_informação}(A)' = \frac{|A_{\text{informado}}|}{|S|}\,[E'(S) - E'(S,A)] \quad (21)$$

Onde:

$|A_{\text{informado}}|$ denota o número de tuplas em S em que o valor de A é informado e E'(S) e E'(S,A) são calculadas usando (18) e (19) somente considerando as tuplas em que o valor de A é informado.

E, neste caso, a razão de ganho passa a ser calculada através de:

$$\text{Razão\_ganho}(A)' = \frac{\text{Ganho\_informação }(A)'}{-\sum_{v \in \text{Domínio}'(A)} \frac{|S_v|}{|S|} \log_2\left(\frac{|S_v|}{|S|}\right)} \quad (22)$$

No denominador de (22), o domínio é alterado para considerar os valores não informados como um grupo adicional.

No caso de observações parciais, a maneira como as partições de treinamento D_j são geradas também é modificada. Como visto anteriormente, uma vez que um teste é escolhido para um nó, as tuplas são particionadas em duas ou mais partições D_j. O algoritmo C4.5 envia as tuplas com valores não informados para o nó para todos os nós filhos, mas com um peso calculado com base na proporção de cada valor do domínio presente em D (*fractional instances*). Por exemplo, considere um nó A_1 que pode assumir dois valores: val_1 e val_2. Se, na partição D, estão 10 tuplas com $A_1 = val_1$, 30 tuplas com $A_1=val_2$ e 5 tuplas com A_1 não informado, então, estas 5 tuplas serão enviadas para as duas partições resultantes mas com pesos multiplicados por 10/40 para a condição $A_1=val_1$ e 30/40 para a condição $A_1=val_2$.

No momento da predição para uma tupla, suponha que um atributo não informado está representado como um nó na árvore de decisão previamente aprendida. Ao chegar a este nó, a tupla será propagada na árvore com pesos proporcionais às frequências dos valores observados para este atributo na base. Quando os múltiplos caminhos alcançam as folhas, as probabilidades das classes presentes nas folhas são combinadas, isto é, ponderadas de acordo com os pesos, de modo que a classe com maior probabilidade resultante é indicada.

Objetivando a prevenção do *overfitting*, as árvores de decisão utilizam poda evitando árvores demasiadamente complexas. A árvore de decisão C4.5 implementada na ferramenta WEKA (também conhecida como J48) possui duas estratégias para poda: pós-poda e pré-poda.

A pós-poda pega uma árvore totalmente crescida e descarta partes da árvore na direção de baixo para cima, ou seja, inicia-se a partir das folhas da árvore completa e atua em direção ao nó raiz. Para decidir o que podar, J48 calcula o erro antes e depois da poda e três métodos de pós-poda podem ser empregados.

No primeiro método de pós-poda, conhecido como substituição de sub-árvore (*subtree replacement*), a substituição dos nós é realizada se o erro estimado para a folha prospectiva não é maior que a soma dos erros estimados para os nós folha atuais da sub-árvore. O segundo método de pós-poda utilizado é conhecido por subida de sub-árvore (*subtree raising***)**, o qual substitui uma sub-árvore pelo seu ramo mais povoado se tal substituição não implicar o crescimento do erro estimado. As Figuras 7 e 8 ilustram respectivamente as podas por substituição e subida de sub-árvore. Tanto *subtree replacement* quanto *subtree raising* são métodos de poda baseados em erro que computam um intervalo de confiança para o erro de classificação sobre os dados de treinamento e usam o limite superior de tal intervalo como uma estimativa para o erro fora da amostra para tomar as decisões de poda (estimativa de erro pessimista). A quantidade de pós-poda é regulada por um intervalo de confiança que é determinado com base em um nível de confiança pré-ajustado para o classificador (hiperparâmetro). Valores mais baixos para o nível de confiança implicam um intervalo de confiança mais largo, o que implica estimativas de erro mais pessimistas produzindo árvores mais podadas.

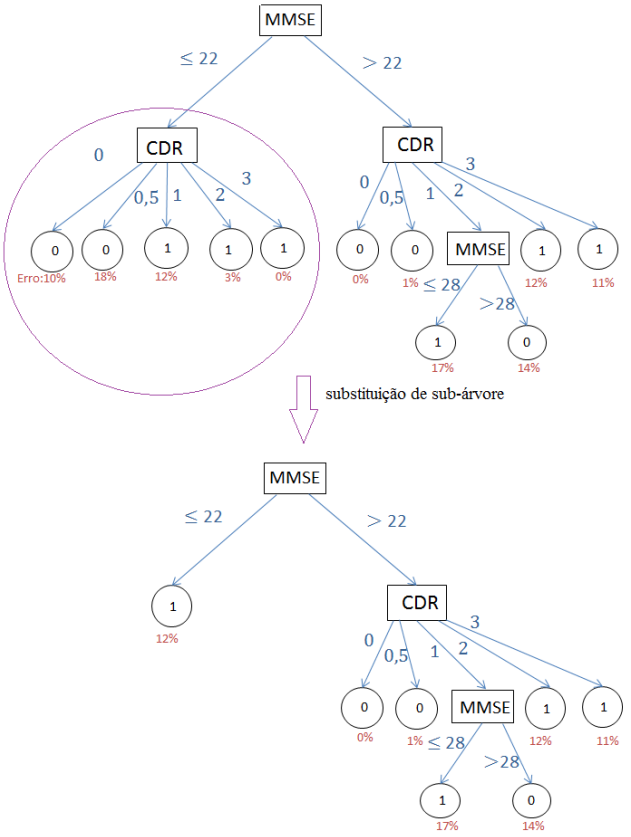


Figura 7: Poda por substituição de sub-árvore.

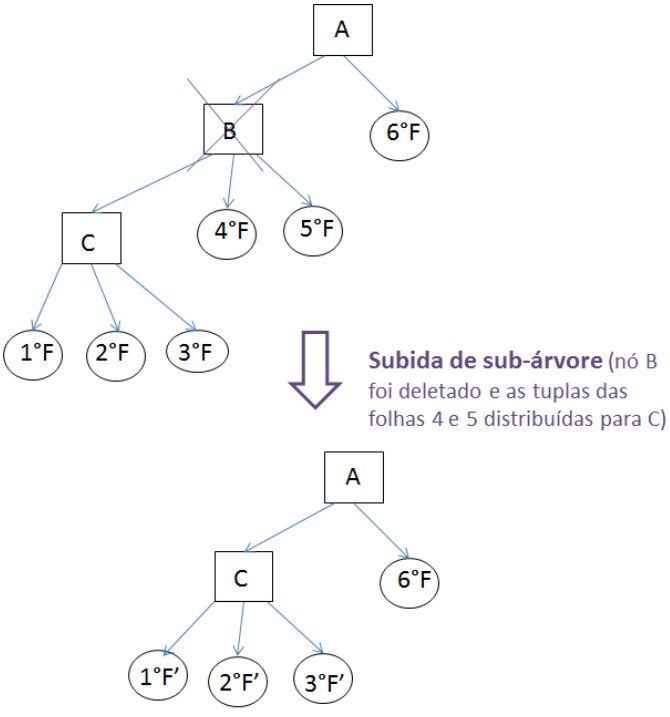


Figura 8: Poda por subida de sub-árvore.

O terceiro método de pós-poda que pode ser utilizado pela árvore J48 é chamado de poda por erro reduzido (*reduced error pruning*), o qual utiliza conjuntos distintos de dados para o crescimento e para a poda da árvore, seguindo os seguintes passos:

1. Particione o conjunto de dados de treinamento em dois conjuntos: um para o crescimento e outro para a poda;
2. Aprenda uma árvore de decisão completa que classifica todos os exemplos no conjunto para crescimento;
3. Enquanto o erro para o conjunto de poda não aumentar:
   3.1. Tente substituir cada nó por uma folha (rotulado com a classe majoritária);
   3.2. Avalie a sub-árvore resultante usando o conjunto para poda;
   3.3. Realize a substituição que resulta na redução de erro máxima;
4. Retorne a árvore de decisão resultante.

Quanto ao passo 1, os tamanhos dos conjuntos de dados são ajustados de acordo com um hiperparâmetro expressando o número total de *folds* a ser utilizado, em que um destes *folds* é utilizado como o conjunto de dados para poda e os demais *folds* como o conjunto de dados para o crescimento.

A árvore J48 pode também utilizar pré-poda, a qual atua sobre a maneira como a árvore cresce de cima para baixo impedindo-a de crescer demasiadamente. A pré-poda é regulada por um hiperparâmetro fixando o número mínimo de tuplas que uma folha deve representar. Durante a indução da árvore de decisão, um novo ramo não é criado a menos que tal ramo contenha pelo menos o número mínimo especificado. Se pelo menos um ramo de um nó resultaria em um nó folha representando tuplas em um número menor que o mínimo, nenhum dos ramos é criado a partir do nó e as tuplas que seguiriam pelos ramos passam a ser representadas pelo nó que passa a ser folha.

3.7 FLORESTA ALEATÓRIA

Floresta Aleatória (*Random Forest* - RF), proposta por BREIMAN (2001), é um modelo composto (*ensemble*), formado a partir da combinação de classificadores (árvores de decisão). RF baseia-se em *bagging* e seleção de atributos aleatória (HAN, KAMBER e PEI, 2011).

Dado um conjunto de dados D, com N tuplas, *bagging* funciona da seguinte forma. Para a iteração i (i = 1, 2,..., I), um conjunto de treinamento, D_i, contendo n tuplas (n<N) é amostrado com substituição do conjunto original D. Uma vez que a amostragem com substituição é utilizada, algumas tuplas de D podem não ser incluídas em nenhum conjunto D_i enquanto outras podem ser incluídas em mais de um conjunto D_i. Então, um modelo M_i, é aprendido para cada conjunto de treinamento D_i. Para classificar uma tupla com classe desconhecida, cada classificador M_i retorna sua predição para a classe que conta como um

voto. O *ensemble* M*, conta os votos e atribui para a tupla de teste o rótulo de classe com maior número de votos.

Para RF, cada classificador no *ensemble* é uma árvore de decisão de forma que a coleção de árvores forma uma "floresta". Como explicado no próximo parágrafo, cada árvore individual é construída usando seleção de atributos aleatória em cada nó para determinar a partição. Durante a predição, cada árvore vota e a classe mais popular é retornada.

Suponha um conjunto de dados de treinamento, D, contendo n tuplas. O procedimento para gerar I árvores de decisão para o ensemble é explicado a seguir. Para cada iteração, i (i = 1, 2,..., I), um conjunto de treinamento D_i com n tuplas é amostrado com substituição a partir de D. Seja K o número de atributos que pode ser utilizado para determinar as partições em cada nó, em que K é menor que o número total de atributos da base de treinamento excluindo-se a classe. Na construção de cada árvore de decisão M_i, seleciona-se aleatoriamente K atributos para compor a lista de atributos candidatos para o nó.

Na implementação presente na WEKA para RF, as árvores podem ser pré-podadas através da fixação de uma profundidade máxima para as árvores da floresta. Além disso, o número de árvores a serem geradas (I), o número de atributos aleatoriamente escolhidos (K) e a profundidade máxima das árvores são hiperparâmetros do classificador.

A implementação presente na WEKA para RF é capaz de lidar com atributos discretos e contínuos e com observações parciais, pois herda tais capacidades da árvore de decisão empregada na construção da floresta, a qual é construída de maneira similar a árvore J48, porém com seleção de atributos aleatória para gerar os nós da árvore.

3.8 K VIZINHOS MAIS PRÓXIMOS

k vizinhos mais próximos (*k Nearest Neighbors* - k-NN), descrito por AHA, KIBLER e ALBERT (1991), é um classificador do tipo *lazy*, o que significa que não há um modelo previamente construído para classificar uma tupla de teste. A predição é realizada usando o conjunto de dados de treinamento diretamente com a vantagem de que, se tal conjunto mudar, não é necessário reconstruir um modelo. Contudo, a predição para o k-NN é realizada a um custo computacional mais alto quando comparado com os classificadores convencionais que constroem um modelo antes da predição (classificadores do tipo *eager*). A predição é realizada por analogia. As classes atribuídas a tuplas similares contidas na base de treinamento determinarão a classe desconhecida para uma tupla de entrada. Considerando que cada tupla possui d atributos, então elas podem ser representadas por um ponto no espaço d-dimensional. O k-NN busca em tal espaço k tuplas mais próximas que são vizinhas da tupla a

ser predita (k vizinhos mais próximos) e atribui a esta tupla a classe majoritária entre tais vizinhos (*voting*). A implementação da ferramenta WEKA para o k-NN, denominada IBk, opcionalmente permite atribuir pesos para as contribuições dos k vizinhos mais próximos na votação, de forma que vizinhos com menor distância para a tupla sendo predita (entre os k mais próximos) podem contribuir mais para a classe atribuída do que vizinhos com maior distância. k é um hiperparâmetro para o k-NN que deve ser ajustado, pois diferentes valores para k conduzem a resultados distintos. A proximidade entre tuplas é definida com base em uma métrica de distância, sendo a distância Euclidiana comumente utilizada, mas outras métricas de distância podem ser empregadas.

Considere duas tuplas $\mathbf{x_1}$= (a_{11},a_{12},...,a_{1d}) e $\mathbf{x_2}$ = (a_{21},a_{22},...,a_{2d}) com atributos contínuos. A distância Euclidiana entre $\mathbf{x}_1$ e $\mathbf{x}_2$ será dada por:

$$\text{dist}(\mathbf{x_1}, \mathbf{x_2}) = \sqrt{\sum_{i=1}^{d}(a_{1i} - a_{2i})^2} \qquad (23)$$

Em que: a_{1i} denota o valor da tupla $\mathbf{x_1}$ para o atributo i e a_{2i} denota o valor da tupla $\mathbf{x_2}$ para o mesmo atributo i.

Quanto menor dist($\mathbf{x_1}$, $\mathbf{x_2}$), mais próximas e similares serão $\mathbf{x_1}$ e $\mathbf{x_2}$.

Tipicamente, os valores de cada atributo são normalizados antes de serem colocados em (23) evitando que atributos com domínios de intervalos grandes tenham peso muito maior que atributos com domínios de intervalos pequenos. A normalização min-max é utilizada para transformar o valor v de um atributo contínuo para v_0 com domínio [0, 1] (HAN, KAMBER e PEI, 2011).

Para calcular a distância levando-se em conta atributos categóricos, IBk usa (23) mas, para atributos categóricos, compara o valor do atributo i na tupla $\mathbf{x_1}$ com o valor do mesmo atributo na tupla $\mathbf{x_2}$: se eles são idênticos, então $a_{1i} - a_{2i}$ é igualado a 0, caso contrário, $a_{1i} - a_{2i}$ é igualado a 1.

IBk trata observações parciais da seguinte maneira: se o valor de certo atributo não está informado na tupla $\mathbf{x_1}$ e/ou na tupla $\mathbf{x_2}$, é assumida a máxima diferença possível. Para atributos categóricos, tal diferença será 1 se os dois valores não forem informados ou se apenas um deles não for informado. Suponha que os atributos contínuos tenham sido normalizados de forma que possam assumir valores no intervalo [0,1]. Se o atributo é contínuo e para ambas as tuplas ele não está informado, tal diferença será 1. Se apenas um valor não for informado e o outro valor estiver presente (denotado como v_0), então tal diferença será max(|1– v_0 |,|0 – v_0|), em que max(val,val') denota o valor máximo entre dois valores val e val'.

IBk permite opcionalmente atribuir pesos para as contribuições dos k vizinhos mais próximos na votação para a predição da classe. No caso em que se opta por utilizar tais pesos, para cada vizinho, pode ser atribuído um peso igual a 1/dist ou 1-dist (conforme hiperparâmetro a ser ajustado) para obtenção da classe mais provável, em que dist é a distância entre o respectivo vizinho e a tupla a ser predita.

3.9 MÁQUINAS DE VETORES DE SUPORTE

Suponha um problema de classificação binária em que os dados são linearmente separáveis. Seja um conjunto de dados D contendo $(\mathbf{x_1}, y_1)$, $(\mathbf{x_2}, y_2)$, ..., $(\mathbf{x_N}, y_N)$, em que $\mathbf{x_n}$ é um vetor representando uma tupla de treinamento associada com um rótulo de classe $y_n \in \{+1, -1\}$ e N é o número de tuplas contidas em D. Para facilitar a visualização, considere que $\mathbf{x_n}$ possui apenas duas dimensões como ilustrado na Figura 9. Como os dados são linearmente separáveis, uma reta é suficiente para separar as tuplas (pontos) que pertencem à classe -1 das tuplas que pertencem à classe +1. Contudo, como ilustrado na Figura 9 (a), existe um número infinito de retas que podem ser usadas para isto. A melhor reta será aquela com erro de classificação mínimo para tuplas ainda não vistas. Generalizando este caso para um número maior de dimensões, é desejável obter o melhor hiperplano que será aquele com a maior margem para os pontos mais próximos uma vez que este provavelmente proverá maior Acc para a classificação de dados futuros do que outros hiperplanos com menor margem (ver Figura 9 (b-c)). Durante o aprendizado, o algoritmo SVM determina o hiperplano com a maior margem, o qual é chamado de hiperplano com margem máxima (*Maximum Marginal Hyperplane* - MMH). A margem associada ao MMH provê a maior separação entre as classes. A distância de um hiperplano para um dos lados de sua margem é igual à distância do hiperplano para o outro lado de sua margem, onde os lados da margem são paralelos ao hiperplano. Para o MMH, os lados da margem sempre passam pelos respectivos pontos de treinamento mais próximos pertencentes a cada classe.

Um hiperplano de separação pode ser escrito como:

$$\mathbf{w} \cdot \mathbf{x_n} + b = 0 \qquad (24)$$

Onde:

- O símbolo "." denota o produto interno entre dois vetores;
- b é um escalar;
- **w** é um vetor de pesos dado por (em que d denota o número de atributos):

$$\mathbf{w}=\begin{bmatrix} w_1 \\ w_2 \\ \dots \\ w_d \end{bmatrix}$$

- $\mathbf{x_n}$ é um vetor com d atributos:

$$\mathbf{x_n}=\begin{bmatrix} a_{n1} \\ a_{n2} \\ \dots \\ a_{nd} \end{bmatrix}$$

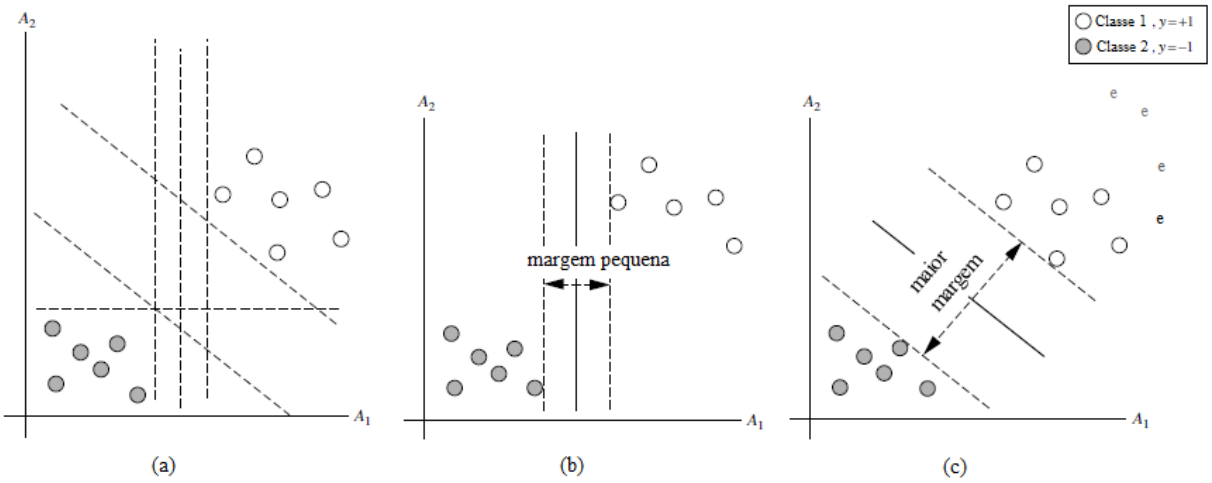


Figura 9: Dados de treinamento 2-D que são linearmente separáveis: (a) retas de separação possíveis representadas por linhas pontilhadas, (b) reta com margem pequena e (c) reta com a maior margem possível. Adaptado de: HAN, KAMBER e PEI, 2011.

Se os dados são linearmente separáveis, **w** e b podem ser ajustados de forma que os hiperplanos definindo os lados da margem sejam escritos como:

$$H1: \mathbf{w} \cdot \mathbf{x_n} + b \geq 1 \qquad \text{para } y_n=+1 \qquad (25)$$

$$H2: \mathbf{w} \cdot \mathbf{x_n} + b \leq -1 \qquad \text{para } y_n=-1 \qquad (26)$$

Qualquer ponto sobre ou acima de H1 pertencerá à classe 1 e qualquer ponto sobre ou abaixo de H2 pertencerá à classe -1.

Combinando (25) e (26), obtém-se:

$$\mathbf{y_n}(\mathbf{w} \cdot \mathbf{x_n} + b) \geq 1, \forall\, n \qquad (27)$$

Os pontos localizados sobre H1 ou H2 são chamados de vetores de suporte e estão igualmente próximos de MMH. Essencialmente, os vetores de suporte são os pontos mais difíceis de classificar e ao mesmo tempo os que fornecem informação mais importante para a classificação definindo o MMH.

A distância entre o hiperplano de separação e qualquer ponto sobre H1 é $1/\|\mathbf{w}\|$ (CORTES e VAPNIK, 1995), onde $\|\mathbf{w}\|$ é a norma Euclidiana de **w**, isto é, $\sqrt{\mathbf{w} \cdot \mathbf{w}}$. Por

definição, esta também é a distância entre qualquer ponto sobre H2 e o hiperplano de separação. Portanto, a margem máxima é dada por:

$$\text{margem\_max} = 2/\|\mathbf{w}\| \quad (28)$$

Sendo assim, deseja-se maximizar (28) sujeito à condição (27), que pode ser reescrito como:

$$\text{Minimizar } 0{,}5\ \mathbf{w}\,\mathbf{w}^{T}, \text{ sujeito a: } y_n(\mathbf{w^T x_n} + b) \geq 1 \text{ para n=1, 2,..., N} \quad (29)$$

Reescrevendo (29) usando a formulação de *Lagrange* como mostrado em CORTES e VAPNIK (1995), o problema torna-se:

$$\text{Minimizar } L(\mathbf{w}, b, \boldsymbol{\alpha}) = 0{,}5\ \mathbf{w}\,\mathbf{w}^{T} - \sum_{n=1}^{N} \alpha_n(y_n(\mathbf{w}^{T}\mathbf{x_n} + b) - 1) \text{ com respeito a } \mathbf{w}, \text{ b e maximizar com respeito a } \boldsymbol{\alpha} \quad (30)$$

Onde $\boldsymbol{\alpha}$ é um vetor de tamanho N contendo os multiplicadores de *Lagrange*, os quais são não negativos.

Derivando $L(\mathbf{w}, b, \alpha)$ com respeito a **w** e b e igualando a 0 conduz a:

$$\mathbf{w} = \sum_{n=1}^{N} \alpha_n y_n \mathbf{x_n} \quad (31)$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \quad (32)$$

Substituindo (31) e (32) em $L(\mathbf{w}, b, \alpha)$, obtém-se:

$$L(\alpha) = \sum_{n=1}^{N} \alpha_i - 0{,}5 \sum_{n=1}^{N} \sum_{m=1}^{N} (y_n\, y_m\ \alpha_n\, \alpha_m\, \mathbf{x_n^T x_m}) \quad (33)$$

Agora, deseja-se maximizar (33) com respeito a $\boldsymbol{\alpha}$ sujeito a $\alpha_n \geq 0$, para n=1,...,N e a $\sum_{n=1}^{N} \alpha_n y_n = 0$, que pode ser reescrito como:

Minimizar $0{,}5 \sum_{n=1}^{N} \sum_{m=1}^{N} (y_n\, y_m\ \alpha_n\, \alpha_m\, \mathbf{x_n^T x_m}) - \sum_{n=1}^{N} \alpha_n$ com respeito a $\boldsymbol{\alpha}$ e sujeito a $\alpha_n \geq 0$, para n=1,...,N e a $\sum_{n=1}^{N} \alpha_n y_n = 0$ (34).

Para bases de treinamento pequenas, por exemplo, contendo em torno de N=2000 tuplas, qualquer pacote de software que resolva problemas de programação quadrática (*Quadratic Programming* - QP) [11] pode ser utilizado para solucionar (34). A implementação nativa da ferramenta WEKA para a solução de (34) é capaz de lidar com bases maiores sendo chamada de Otimização Mínima Sequencial (*Sequential Minimal Optimization* - SMO), a qual decompõe o problema QP em problemas QP menores e mais fáceis de serem solucionados (PLATT, 1999).

A solução de (34) retorna os α_i e os substituindo em (31) obtém-se o **w** aprendido a partir da base de treinamento. A otimização pela formulação de *Lagrange* usa a condição *Karush-Kuhn-Tucker* (KKT) $\alpha_n(y_n(\mathbf{w}^{T}\mathbf{x_n} + b) - 1) = 0$ (CORTES e VAPNIK, 1995), de

[11] Problemas em que se deseja maximizar ou minimizar uma função objetiva quadrática sujeito a um conjunto de restrições lineares.

forma que $\alpha_n \neq 0$ somente para os vetores de suporte, pois estes estão sobre os hiperplanos definindo os lados da margem, ou seja, onde $y_n(\mathbf{w}^T\mathbf{x_n} + b) = 1$. Para pontos interiores, vale $y_n(\mathbf{w}^T\mathbf{x_n} + b) > 1$, resultando em $\alpha_n = 0$. Sendo assim, $\boldsymbol{\alpha}$ é um vetor esparso que contém valores distintos de zero somente para os vetores de suporte. Para obter o valor de b, qualquer vetor de suporte pode ser utilizado bastando substituir um $\mathbf{x_n}$ e y_n (correspondentes a qualquer $\alpha_n > 0$) e o $\mathbf{w}$ aprendido em $y_n(\mathbf{w}^T\mathbf{x_n} + b) = 1$. Na predição, a classe de uma instância de teste $\mathbf{x_t}$ será:

$$y_t = \text{sgn}[\sum_{n=1}^{L}(y_n\alpha_n\mathbf{x_n} \,.\, \mathbf{x_t} + b)] \quad (35)$$

Onde: sgn[valor] retorna 1 se valor >1 ou -1 caso contrário; L é o número de vetores de suporte; y_n é o valor da classe para o vetor de suporte $\mathbf{x_n}$; e α_n e b são parâmetros numéricos determinados pelo aprendizado, isto é, pela otimização explicada anteriormente.

SVM também pode ser aplicado para dados de treinamento que não sejam linearmente separáveis, mas, neste caso, é necessário transformar o espaço original dos dados de entrada em um espaço de dimensão superior usando mapeamento não linear. Uma vez que os dados sejam transformados de acordo com este novo espaço, o processo de aprendizado determina um hiperplano de separação linear neste novo espaço, recaindo em um problema QP, que pode ser resolvido como anteriormente explicado, ou seja, baseando-se no SVM para dados linearmente separáveis. O MMH obtido neste novo espaço corresponde a uma hipersuperfície de separação não linear no espaço original.

Note que (35), assim como os dados que são passados para obter $\boldsymbol{\alpha}$ usando (33), contém produtos internos entre vetores **x.** Sendo assim, na solução do problema QP no novo espaço de dimensão superior, as tuplas de treinamento aparecerão somente na forma de produtos internos, $\varphi(\mathbf{x_i})\cdot\varphi(\mathbf{x_j})$, em que $\varphi(\mathbf{x})$ é uma função de mapeamento não linear aplicada para transformar as tuplas de treinamento. Calcular o produto interno entre as tuplas de dados transformadas é matematicamente equivalente a aplicar uma função núcleo (*kernel function*), $K(\mathbf{x_i}, \mathbf{x_j})$ aos dados de entrada originais, em que:

$$K(\mathbf{x_i}, \mathbf{x_j}) = \varphi(\mathbf{x_i})\cdot\varphi(\mathbf{x_j}) \quad (36)$$

Toda vez que $\varphi(\mathbf{x_i})\cdot\varphi(\mathbf{x_j})$ aparecer no treinamento ou na predição, este será substituído por $K(\mathbf{x_i}, \mathbf{x_j})$, de forma que os cálculos são simplificados, pois são realizados sobre os dados do espaço de entrada original de menor dimensão.

Algumas funções núcleo disponíveis na WEKA são:

- Função polinomial de grau E:

$$Kp\,(\mathbf{x_i},\ \mathbf{x_j},\ E) = (\mathbf{x_i}\,.\,\mathbf{x_j} + 1)^E \quad (37)$$

- Função polinomial normalizada de grau E:

$$K_{NP}(\mathbf{x_i}, \mathbf{x_j}, E) = \frac{Kp\ (\mathbf{x_i},\ \mathbf{x_j},\ E)}{\sqrt{Kp\ (\mathbf{x_i},\ \mathbf{x_i},\ E)\ Kp\ (\mathbf{x_j},\ \mathbf{x_j},\ E)}} \quad (38)$$

- RBF com hiperparâmetro gama γ:

$$K_{RBF}(\mathbf{x_i}, \mathbf{x_j}, \gamma) = e^{[-\gamma \|\mathbf{x_i}-\mathbf{x_j}\|^2])} \quad (39)$$

- *Kernel* universal baseado em função de Pearson VII com hiperparâmetros sigma σ e ômega ω (ÜSTÜN, MELSSEN e BUYDENS, 2006):

$$K_{PUK}(\mathbf{x_i}, \mathbf{x_j}, \sigma, \omega) = \frac{1}{\left[1+\left(\frac{2\sqrt{\|\mathbf{x_i}-\mathbf{x_j}\|^2}\sqrt{2^{(1/\omega)}-1}}{\sigma}\right)^2\right]^{\omega}} \quad (40)$$

Cada uma destas funções núcleo resulta em um classificador não linear distinto para o espaço de entrada original e possui diferentes hiperparâmetros a serem ajustados (E, γ, σ e ω).

Objetivando prevenir o *overfitting*, SVM foi modificado dando origem ao SVM com margem suave (*soft margin SVM* - sm-SVM) o qual permite violações da margem (alguns pontos desrespeitando a margem) de forma que $y_n(\mathbf{w^T x_n} + b) \geq 1$ pode falhar por uma folga ξ_n (ver Figura 10), onde $\xi_n \geq 0$, de forma que:

$$y_n(\mathbf{w^T x_n} + b) \geq 1 - \xi_n \quad (41)$$

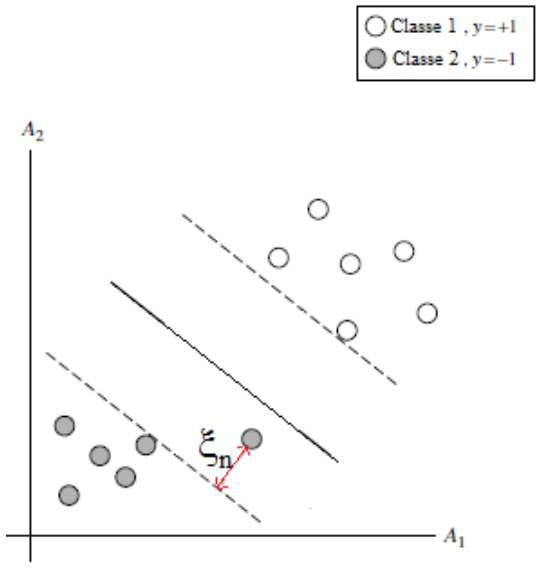


Figura 10: Violação de margem no SVM de margem suave.

A violação total, a qual representa uma medida de erro de classificação dentro da amostra, será dada por:

$$\text{Violação\_total\_margem} = \sum_{n=1}^{N} \xi_n \quad (42)$$

Algum erro dentro da amostra pode conduzir a uma melhor generalização fora da amostra e, por causa disto, sm-SVM foi criado buscando:

$$\text{Minimizar } 0{,}5\ \mathbf{w}\,\mathbf{w}^{T} + C\sum_{n=1}^{N}\xi_n, \text{ sujeito a:}$$

$$y_n(\mathbf{w}^{T}\mathbf{x_n} + b)\ \geq 1-\xi_n \text{ e } \xi_n \geq 0, \text{ para n=1,2,..., N} \quad (43)$$

Em (43), C é uma constante positiva que regula o quanto a margem pode ser violada. Valores maiores para C resultam em ξ_n menores (menor violação da margem), de modo que, quando C tende a infinito, recai-se no problema anteriormente descrito, isto é, SVM sem margem suave (*hard margin* SVM). Por outro lado, valores menores para C resultam em margens mais largas com violações de margem mais frequentes. Reescrever (43) usando a formulação de *Lagrange* conduz a:

Minimizar $0{,}5\sum_{n=1}^{N}\sum_{m=1}^{N}(y_n\, y_m\ \alpha_n\, \alpha_m\, \mathbf{x_n^T x_m}) - \sum_{n=1}^{N}\alpha_n$ com respeito a $\boldsymbol{\alpha}$ e sujeito a $0 \leq \alpha_n \leq C$ (para n= 1,2,...,N) e a $\sum_{n=0}^{N}\alpha_n y_n = 0$ (44)

Fornecendo-se os dados de treinamento e fixando um valor para C (hiperparâmetro do sm-SVM), o problema de otimização expresso em (44) pode ser resolvido por QP para obtenção do vetor $\boldsymbol{\alpha}$. Cada vetor de suporte ($\mathbf{x_n}$) localizado sobre os lados da margem corresponde a um $\boldsymbol{\alpha_n}$ no intervalo (0,C) possuindo ξ_n=0 e cada vetor de suporte ($\mathbf{x_n}$) violando a margem corresponde a um $\boldsymbol{\alpha_n}$=C possuindo ξ_n>0. A partir de todos os vetores de suporte, **w** e b são obtidos como explicado anteriormente, completando o aprendizado. A partir de então, a classe para uma instância de teste pode ser predita utilizando (35).

Em (35), a saída da predição será o valor da classe com probabilidade igual a 100%. A implementação SMO da WEKA para sm-SVM permite ajustar a saída da predição para retornar uma probabilidade para a classe predita variando entre 50% e 100% (saída soft). Para isto, o SMO deve ser ajustado com o hiperparâmetro –M. Neste caso, $\sum_{n=1}^{L}(y_n\alpha_n\mathbf{x_n}\,.\,\mathbf{x_t} + b)$ é modelado como um preditor de um modelo RL e a saída deste modelo retorna a probabilidade. Na presente tese, sm-SVM de saída soft (SMO ajustado com o hiperparâmetro –M) pode ser um dos classificadores escolhidos pelo processo de aprendizado. Cabe ressaltar ainda que a implementação SMO da WEKA possui os seguintes pré-processamentos nela embutidos: substituição de dados não informados pela média (para atributos contínuos) ou moda (para atributos discretos); transformação de cada atributo categórico com k valores distintos em k-1 atributos booleanos; e normalização de todos os atributos contínuos usando a normalização min-max para mapeá-los para o domínio [0, 1]. Um hiperparâmetro regula tal normalização que pode ser desabilitada ou substituída por padronização dos dados de

treinamento, isto é, redimensionamento dos atributos contínuos para que eles tenham μ igual a 0 e σ igual a 1 assumindo que os mesmos possuam distribuições Gaussianas.

3.10 A MANEIRA DE LIDAR COM OBSERVAÇÕES PARCIAIS DE CADA CLASSIFICADOR

Os classificadores das Seções 3.2 a 3.9 lidam de maneiras diferentes com observações parciais.

BN, para o cálculo da probabilidade de uma tupla de teste pertencer a uma dada classe, na inferência Bayesiana, atribui as evidências apenas para os valores observados (informados). No aprendizado de parâmetros, BN usa o algoritmo EM.

NB, para o cálculo da probabilidade da tupla de teste pertencer a uma dada classe, apenas leva em consideração os valores observados a_i para determinar o valor da expressão (8). A estimativa $\widehat{P}(a_i|y)$ é calculada mesmo se o atributo não estiver informado em todas as tuplas da base de treinamento, considerando-se apenas as tuplas em que o atributo estiver informado.

Para o cálculo da probabilidade da tupla de teste pertencer a uma dada classe, A1DE faz a média dos estimadores de uma dependência considerando os "super pais" correspondentes aos valores informados na tupla de teste, se estes valores de "super pais" estão informados na base de treinamento. Caso contrário, A1DE usa a estimativa NB.

RL-NB, na parte RL do modelo, substitui valores não informados pela média (se o atributo for contínuo) ou pela moda (se o atributo for discreto) e, na parte NB do modelo, trata as observações parciais como o NB trata.

Árvore de Decisão C4.5, na indução da árvore, as partições são criadas em cada nó da árvore utilizando o conceito de *"fractional instances"* como descrito na Seção 3.6. Na predição, se um atributo não informado está representado como um nó na árvore de decisão, a tupla de teste será propagada na árvore com pesos proporcionais às frequências dos valores informados para este atributo na base de treinamento. Desta forma, múltiplos caminhos alcançam as folhas e as probabilidades das classes presentes nas folhas são ponderadas para compor a probabilidade final.

RF lida com observações parciais da mesma maneira que a árvore de Decisão C4.5 para as árvores da floresta.

k-NN, no cálculo da distância entre duas tuplas, se o valor de certo atributo não está informado em pelo menos uma das tuplas, é assumida a máxima diferença possível para este atributo.

SVM, na implementação SMO, substitui dos dados não informados pela média (para atributos contínuos) ou moda (para atributos discretos).

3.11 MÉTODOS PARA AVALIAÇÃO DE CLASSIFICADORES E MEDIDAS DE DESEMPENHO

Na seleção de um algoritmo de classificação para uma base de dados, é desejável comparar o seu desempenho com outros algoritmos de classificação. Como visto na Seção 3.1, no treinamento, um algoritmo de classificação busca dentro de seu conjunto de hipóteses aquela que minimiza alguma medida de erro dentro da amostra. Após o treinamento, o classificador será usado para predizer o valor de classe para tuplas com classe desconhecida e, portanto, as tuplas de treinamento não podem ser usadas para testar um classificador aprendido a partir destas mesmas tuplas, pois isso conduziria a estimativas otimistas de desempenho. Por isso, em geral, as bases de treinamento e teste devem ser distintas e, como se trata de aprendizado supervisionado e se deseja medir o desempenho do classificador na predição das tuplas da base de teste, os rótulos de classe devem ser conhecidos para as tuplas de ambas as bases.

Os métodos de avaliação de classificadores comumente utilizados na literatura tratam de como particionar a base total originando as bases para treinamento e teste.

No método *holdout*, a base é aleatoriamente particionada em dois conjuntos independentes: um para treinamento e outro para teste. Tipicamente o conjunto de treinamento inclui 2/3 das tuplas e o de teste 1/3 das tuplas. O modelo é aprendido a partir do conjunto de treinamento e, então, uma medida de desempenho é calculada a partir das predições realizadas tendo como entrada as tuplas de teste.

No método de subamostragem aleatória (*random subsampling*), o *holdout* é repetido k vezes e a estimativa para o desempenho é calculada como uma média do valor de desempenho obtido em cada iteração.

No método de validação cruzada com k *folds* (*k-fold* CV), a base total é aleatoriamente particionada em k partes de mesmo tamanho aproximadamente (k *folds*): D_1, D_2,... , D_k. Treinamento e teste são realizados k vezes sendo que em cada iteração i, Di é reservado para teste e os *folds* restantes são reservados para treinamento. Cada tupla da base total é usada o mesmo número de vezes para treinamento (k-1) e uma única vez para teste.

Na validação cruzada do tipo estratificada, o particionamento da base nos *folds* é realizado de forma a preservar a distribuição de classes da base original. Isto é particularmente importante quando a informação sobre a distribuição de classes for usada em

algum pré-processamento da base de treinamento como, por exemplo, em técnicas de balanceamento.

O método *leave-one-out* é um caso especial de *k-fold* CV em que o número de tuplas na base total é atribuído a k, de forma que, em cada iteração, uma tupla é deixada de fora para o teste enquanto as outras são usadas para treinamento. Este método é recomendado para bases pequenas. Para bases maiores, recomenda-se 10-*fold* CV.

Quanto às medidas de desempenho comumente utilizadas, a acurácia (Acc) é a mais intuitiva medindo a proporção de acertos do classificador. A seguir, é mostrado como são calculadas algumas medidas de desempenho comumente utilizadas na avaliação de classificadores binários.

A Acc é dada por:

$$\text{Acc} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \quad (45)$$

A precisão (*precision*) também chamada de valor preditivo positivo (ou *Positive Predictive Value* - PPV) é dada por:

$$\text{Precisão (PPV)} = \frac{\text{TP}}{\text{TP+FP}} \quad (46)$$

Por sua vez, a sensibilidade (*recall*), também chamada de taxa de positivos verdadeiros (ou *True Positive Rate* – TPR) é dada por:

$$\text{Sensibilidade (TPR)} = \frac{\text{TP}}{\text{TP+FN}} \quad (47)$$

A TFP (ou *False Positive Rate* – FPR) é dada por:

$$\text{FPR} = \frac{\text{FP}}{\text{FP+TN}} \quad (48)$$

O F1-score, também chamado de *F-measure* é dado por:

$$\text{F}_1 = 2\,\frac{\text{precisão .sensibilidade}}{\text{precisão+sensibilidade}} \quad (49)$$

Em (45-49), TP (*True Positive)* denota o número de instâncias positivas classificadas corretamente como positivas; TN (*True Negative*) denota o número de instâncias negativas classificadas corretamente como negativas; FP (*False Positive*) denota o número de instâncias negativas classificadas erroneamente como positivas; e FN (*False Negative*) denota o número de instâncias positivas classificadas erroneamente como negativas.

A curva ROC (*Receiver Operating Characteristic* curve) é um gráfico que ilustra a habilidade diagnóstica de um classificador binário na medida em que seu limiar discriminativo varia. Ela consiste na curva TPR (47) versus FPR (48). A Figura 11 ilustra

exemplos de curvas ROC obtidas por classificadores distintos. O melhor método de predição possível possui uma curva ROC que passa pelo ponto (0,1), representando 100% de TPR e 0% de FPR e possuindo valor 1 para a Área sob a curva ROC (AUC). Um classificador aleatório gera uma curva ROC representada pela linha diagonal (valor 0,5 para AUC) e curvas acima da diagonal representam classificadores melhores que o aleatório (0,5<AUC<1).

Os classificadores binários comumente classificam uma tupla como positiva se $P(Y=1|\mathbf{x}) > 0{,}5$, em que a variável aleatória Y representa a classe da doença (ou transtorno) a diagnosticar que pode assumir valores $\in \{0,1\}$, em que 0 representa um diagnóstico negativo e 1 um diagnóstico positivo e **x** representa um vetor contendo o conjunto de valores presentes nos atributos.

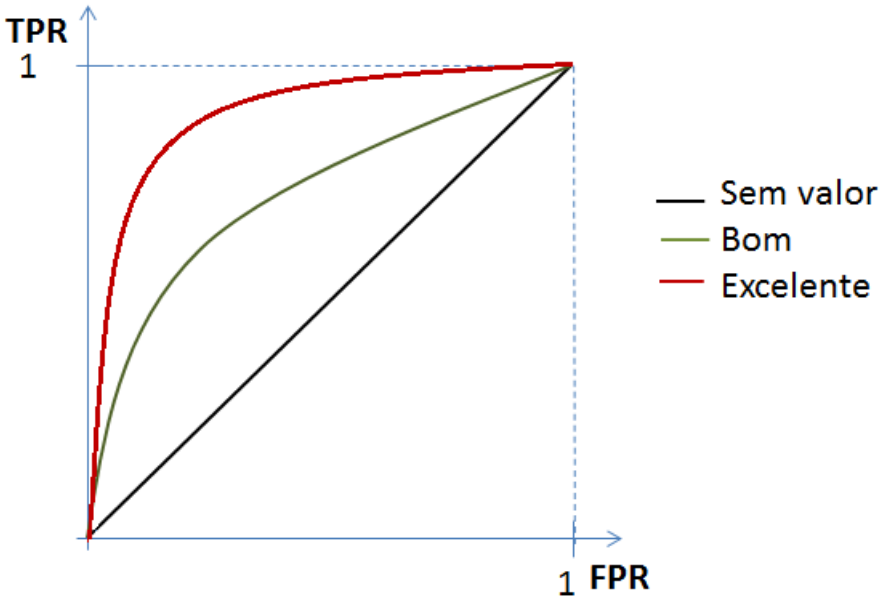


Figura 11: Exemplos de curvas ROC para classificadores distintos.

Seja y_t o valor da classe para uma dada tupla de teste $\mathbf{x}_t$ de acordo com o *Ground Truth*, isto é, o valor verdadeiro da classe para a tupla de teste, e seja $P(Y_t=1|\mathbf{x}_t)$ a probabilidade de y_t ser 1 de acordo com a predição do classificador. Então, o *Root Mean Squared Error* (RMSE) no sentido probabilístico para uma base de dados de teste com T tuplas será dado por:

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}[y_t - P(Y_t = 1|\mathbf{x_t})]^2} \qquad (50)$$

A Tabela 6 resume algumas medidas de desempenho utilizadas na avaliação de classificadores.

Tabela 6: Medidas de desempenho para avaliação de classificadores.

Medida	Acrônimo	Domínio	Melhor pontuação
Acurácia	Acc	[0,1]	1
F1-score	F1	[0,1]	1
Área sob a curva ROC	AUC	[0,1]	1
Raiz quadrada do erro probabilístico quadrático médio	RMSE	[0,1]	0

CAPÍTULO 4 – MODELO DE DECISÃO DINÂMICO

Este capítulo descreve o modelo de decisão proposto para auxiliar os médicos em suas decisões diagnósticas e também descreve o processo de Aprendizado de Máquina Supervisionado (AMS) que constrói tal modelo de decisão, tornando-o dinâmico.

4.1. O MODELO DE DECISÃO DINÂMICO

A aplicação do modelo de decisão proposto a um CDSS em particular visa dar suporte a um processo de diagnóstico específico, seguindo diretrizes clínicas. Estas diretrizes devem ser levantadas junto a médicos especialistas para a definição de como o modelo de decisão será aplicado de acordo com as doenças/transtornos a diagnosticar. O modelo poderá auxiliar os médicos a seguirem o processo de diagnóstico especificado, podendo ser composto por um ou mais classificadores. Ressalte-se que o modelo de decisão dinâmico proposto é restrito a sistemas em que os tipos de dados usados na predição do diagnóstico possam ser representados por atributos contínuos ou discretos.

O modelo de decisão dinâmico proposto nesta tese permite:

- atualizar o classificador utilizado, que deve ser capaz de lidar com observações parciais sobre os casos clínicos disponíveis;
- modificar o conjunto de tipos de dados sobre os casos clínicos disponibilizados pela equipe médica, incluindo novos exames, para melhoria do modelo;
- realizar o balanceamento da base de dados, quando necessário;
- realizar a discretização de valores de atributos contínuos, verificando se esta discretização melhora o desempenho do modelo de decisão.

O que torna dinâmico o modelo de decisão proposto nesta tese é o processo automático de AMS que o constrói, detalhado a seguir.

O modelo de decisão deve ser inicialmente construído utilizando a base de casos clínicos disponível previamente. Para a construção do modelo de decisão inicial, pode ser executado o mesmo processo utilizado para retreinar o modelo de decisão. Quando mais casos clínicos tornam-se disponíveis, o modelo de decisão pode ser retreinado pela execução do processo de aprendizado ilustrado pela Figura 12. Este processo deve ser executado periodicamente para cada doença (ou transtorno) a diagnosticar. Primeiramente, o processo verifica se há a necessidade de construção/retreinamento do modelo. Se o resultado for negativo, o processo é finalizado. Caso contrário, o processo seleciona os dados de saúde de

todos os pacientes diagnosticados para a doença (ou transtorno) em questão, originando um conjunto inicial de dados de treinamento.

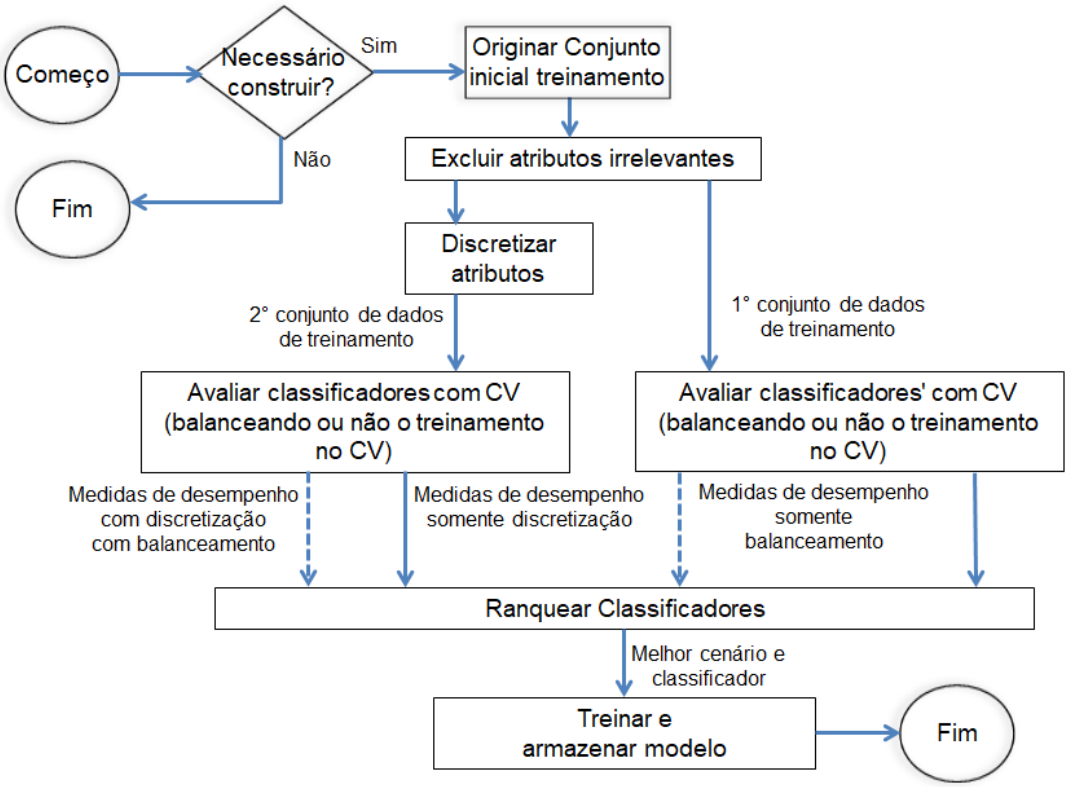


Figura 12: Processo automático de AMS que constrói o modelo de decisão dinâmico proposto.

A condição para verificar a necessidade de construção/retreinamento do modelo depende da implementação do modelo dinâmico. Por exemplo, o processo pode verificar se algum diagnóstico novo para a doença (ou transtorno) em questão foi informado desde o último aprendizado realizado para esta doença (ou transtorno) e retreinar o modelo apenas quando o resultado desta verificação for positivo. Outra alternativa é o processo estabelecer um limite para o número de discordâncias entre os diagnósticos sugeridos pelo sistema e os diagnósticos fornecidos por especialistas, disparando o retreinamento caso tal limite seja excedido.

Caso haja a necessidade de construção do modelo, utilizando o conjunto inicial de dados, as seguintes etapas são realizadas:

1. Exclusão dos atributos com percentual de dados não informados superior a 70% e com valor de ganho de informação inferior a 0.0001 (SEIXAS et al., 2014), originando um primeiro conjunto modificado de dados de treinamento (denotado como 1° conjunto de treinamento). Como visto anteriormente, o valor do ganho de informação expressa o quanto um atributo é eficiente em repartir o conjunto de

dados de treinamento levando em conta os rótulos das classes de forma a reduzir a entropia do conjunto de dados (HAN, KAMBER e PEI, 2011).

2. Discretização de atributos contínuos originando um segundo conjunto modificado de dados de treinamento (denotado como 2° conjunto de treinamento).
3. Avaliação de um conjunto de classificadores capazes de lidar com observações parciais usando validação cruzada (CV):
    a. Se o conjunto de dados inicial estiver balanceado, dois cenários de pré-processamento são avaliados por CV:
        i. Com discretização (usando o 2° conjunto de treinamento),
        ii. Sem discretização (usando 1° conjunto de treinamento)
    b. Se o conjunto de dados estiver desbalanceado, quatro cenários de pré-processamento são avaliados aplicando ou não sobreamostragem somente sobre o treinamento no CV:
        i. Com discretização (usando o 2° conjunto de treinamento) e com sobreamostragem,
        ii. Com discretização (usando 2° conjunto de treinamento) e sem sobreamostragem,
        iii. Sem discretização (usando o 1° conjunto de treinamento) e com sobreamostragem,
        iv. Sem discretização (usando o 1° conjunto de treinamento) e sem sobreamostragem.

    A avaliação considera mais de uma medida de desempenho. Todos os classificadores são ranqueados de acordo com estas medidas e o melhor cenário de pré-processamento/classificador considerando o conjunto de todas as medidas é escolhido. O processo de ranqueamento para a escolha do melhor cenário/classificador está detalhado na Seção 4.1.1.
4. Finalmente, o melhor classificador é treinado de acordo com o melhor cenário e o modelo é armazenado.

Na Etapa 2, é aplicada a discretização supervisionada proposta por KONONENKO (1995). A aplicação da discretização supervisionada dos atributos contínuos é realizada para avaliar se tal discretização contribuirá ou não para a melhoria do desempenho estimado na classificação. Para a aplicação desta discretização, assume-se que os conjuntos de dados de treinamento e teste possuem a mesma distribuição de probabilidade. Portanto, a discretização

é aplicada sobre o conjunto inteiro de dados antes de se realizar a próxima etapa envolvendo CV.

Na Etapa 3, para os cenários com discretização, os seguintes classificadores são avaliados: RB modelada como proposto por SEIXAS et al. (2014); NB; A1DE; árvore de decisão C4.5; Floresta Aleatória (RF); k-NN; e SVM com margem suave e saída *soft*. Para os cenários sem discretização, os classificadores avaliados são: o modelo híbrido LR–NB; NB; árvore de decisão C4.5; RF, k-NN; e SVM com margem suave e saída *soft*. Alguns dos classificadores avaliados podem ter seus hiperparâmetros ajustados automaticamente de acordo com a implementação realizada. A lista dos hiperparâmetros ajustados nesta tese está apresentada na Tabela 7, assim como os domínios e intervalos avaliados no ajuste destes hiperparâmetros.

Nas Etapas 3-a e 3-b, o conjunto de dados é considerado desbalanceado se ele possui a quantidade de casos majoritários (de tuplas rotuladas com a classe com a maior quantidade de exemplos) igual a pelo menos duas vezes a quantidade de casos minoritários (de tuplas rotuladas com a classe com a menor quantidade de exemplos). Para dobrar a quantidade de casos minoritários, é aplicada uma técnica chamada *Synthetic Minority Oversampling Technique* (SMOTE[12]) (CHAWLA et al., 2002) sobreamostrando os casos minoritários em 100%.

Na Etapa 4, o classificador treinado é armazenado. O conjunto de dados utilizado no treinamento também é armazenado para possível uso pelo sistema que implementar o modelo de decisão dinâmico.

O processo automático de aprendizado incorpora ao modelo de decisão novos exames recém-incluídos pelo centro de saúde, desde que estes possuam dados suficientemente informados e não possuam ganho de informação desprezível.

Além disso, o tipo de CV utilizada na avaliação dos classificadores pode variar de acordo com a implementação. Por exemplo, pode ser utilizado *leave-one-out* ou 10-*fold* CV na forma estratificada (por causa da criação de cenários de pré-processamento com aplicação de sobreamostragem sobre o treinamento).

As medidas de desempenho escolhidas para avaliar os classificadores também dependem da aplicação do modelo dinâmico, pois, de acordo com o CDSS que o utiliza, diferentes medidas de desempenho podem ser adotadas.

[12] Nesta técnica de sobreamostragem, o conjunto de casos minoritários é utilizado para a criação de instâncias sintéticas similares (próximas em distância) que são adicionadas ao conjunto de casos minoritários.

Tabela 7: Lista de hiperparâmetros ajustados na avaliação dos classificadores.

Classificador	Descrição dos hiperparâmetros
Árvore de decisão C4.5 (nome na implementação da WEKA: J48)	-B: usar partições binárias nos atributos categóricos na construção das árvores. -U: não usar pós-poda. -C <nível de confiança na pós-poda>: ajusta o limite de confiança para a poda. Valores menores acarretam em mais poda. Pode assumir valores reais no intervalo de 0 a 1. -S: não realizar subida de sub-árvore, usar substituição de sub-árvore. -M <número mínimo de tuplas>: ajusta o número mínimo de tuplas por folha. Pode assumir valores inteiros no intervalo de 1 a 64.
RF	-I <número de árvores> número de árvores. Pode assumir valores inteiros no intervalo de 2 a 256. -K <número de atributos> número de atributos escolhidos aleatoriamente durante o crescimento da árvore para serem candidatos a nó. Se for igual a 0, este número é ajustado para int ($\log_2$(número total de atributos + 1)), onde int (valor) significa o arredondamento do valor para o inteiro mais próximo. Pode assumir valores inteiros no intervalo de 0 a 32. -depth <profundidade máxima>: ajusta a profundidade máxima das árvores. Se igual a 0, a profundidade não é limitada. Pode assumir valores inteiros no intervalo de 0 a 20. -P <tamanho de bag>: tamanho de cada bag como um percentual do conjunto de treinamento. Fixo em 100.
k-NN (nome na implementação WEKA: IBk)	-k <número de vizinhos>: o número de vizinhos mais próximos a utilizar. Pode assumir valores inteiros no intervalo de 1a 64. -A <algoritmo de busca dos vizinhos mais próximos>: ajusta o algoritmo de busca dos vizinhos mais próximos. Fixo em: busca linear com distância Euclidiana e normalização dos atributos -I <ponderação de distância>: usar ponderação de distância 1/dist para a eleição da classe. -F<ponderação de distância>: usar ponderação de distância 1-dist para a eleição da classe. Se -F e -I não estiverem presentes, nenhuma ponderação é feita na eleição da classe.
sm-SVM (nome na implementação WEKA: SMO)	-C <parâmetro de complexidade>: Ajusta a regularização para o SVM de margem suave. Menores valores para C implicam em mais regularização, isto é, mais violações da margem. Pode assumir valores reais no intervalo de 0,5 a 1,5. -N <0>: normalização dos dados de treinamento, isto é, redimensionamento dos atributos contínuos para o intervalo de 0 a 1. -N <1>: padroniza os dados de treinamento, isto é, redimensiona os atributos contínuos para que eles tenham um μ igual a 0 e um σ igual a 1. Assume distribuições Gaussianas. -N <2>: sem normalização ou padronização dos atributos contínuos. -k <função núcleo> (padrão PolyKernel -E 1). O núcleo pode ser ajustado para: - Polinomial (PolyKernel) com parâmetros -E <expoente> e -L (use termos de ordem mais baixa). O expoente –E pode assumir valores reais no intervalo 0,2 a 5. - Polinomial Normalizado (NormalizedPolyKernel) com parâmetros -E <expoente> e -L(use termos de ordem mais baixa). O expoente -E pode assumir valores reais no intervalo 0,2 a 5. - RBF (RBFKernel) com parâmetro -G <gama>, que pode assumir valores reais no intervalo 10^{-4} a 1. - Kernel universal baseado em função de Pearson VII (Puk) com parâmetros -S <sigma> e -O <ômega>. -S pode assumir valores reais no intervalo de 0,1 a 10 e -O pode assumir valores reais no intervalo de 0,1 a 1.

4.1.1. PROCESSO DE RANQUEAMENTO E ESCOLHA DO MELHOR CENÁRIO DE PRÉ-PROCESSAMENTO E CLASSIFICADOR

O processo de ranqueamento dos classificadores para a escolha do melhor cenário/classificador ocorre em dois níveis:

1. No primeiro nível, para cada cenário:
    a. É calculada a posição (pontuação) de cada classificador de acordo com cada medida (valores de desempenho, por exemplo, Acc, F1, AUC e RMSE), levando-se em consideração casos de empate.
    b. Para cada classificador, são somadas as pontuações relativas a cada medida e o classificador com menor soma é escolhido como o de melhor desempenho.
2. No segundo nível:
    a. Considerando os melhores classificadores de cada cenário, as etapas A e B são realizadas para a escolha do melhor cenário-classificador.

Por exemplo, considere o exemplo de ranqueamento de classificadores ilustrado pela Figura 13, em que dois cenários de pré-processamento são avaliados: com discretização e sem discretização. Para estes cenários, considerando-se a Acc, os valores são colocados em ordem decrescente e uma pontuação para cada classificador é determinada de acordo com a posição do classificador, em que uma menor pontuação representa uma melhor posição em relação aos demais classificadores. O mesmo é feito para as medidas F1 e AUC. Para a medida RMSE, os valores são colocados em ordem crescente, pois quanto menor o valor do RMSE melhor o desempenho, e, a seguir, uma pontuação para cada classificador é determinada de acordo com a posição do classificador para os valores de RMSE. Como ilustrado na Figura 13, o cálculo das pontuações leva em consideração os casos de empate. Para finalizar o ranqueamento em cada cenário, as pontuações de cada classificador, referentes a todas as medidas, são somadas e é escolhido o classificador de menor pontuação total, pois este obteve o melhor desempenho considerando-se o conjunto total das medidas.

A seguir, o mesmo processo de ranqueamento é realizado para os classificadores vencedores dos cenários com discretização e sem discretização para escolher o cenário/classificador de melhor desempenho. No exemplo, foi escolhido o cenário com discretização e o classificador 2.

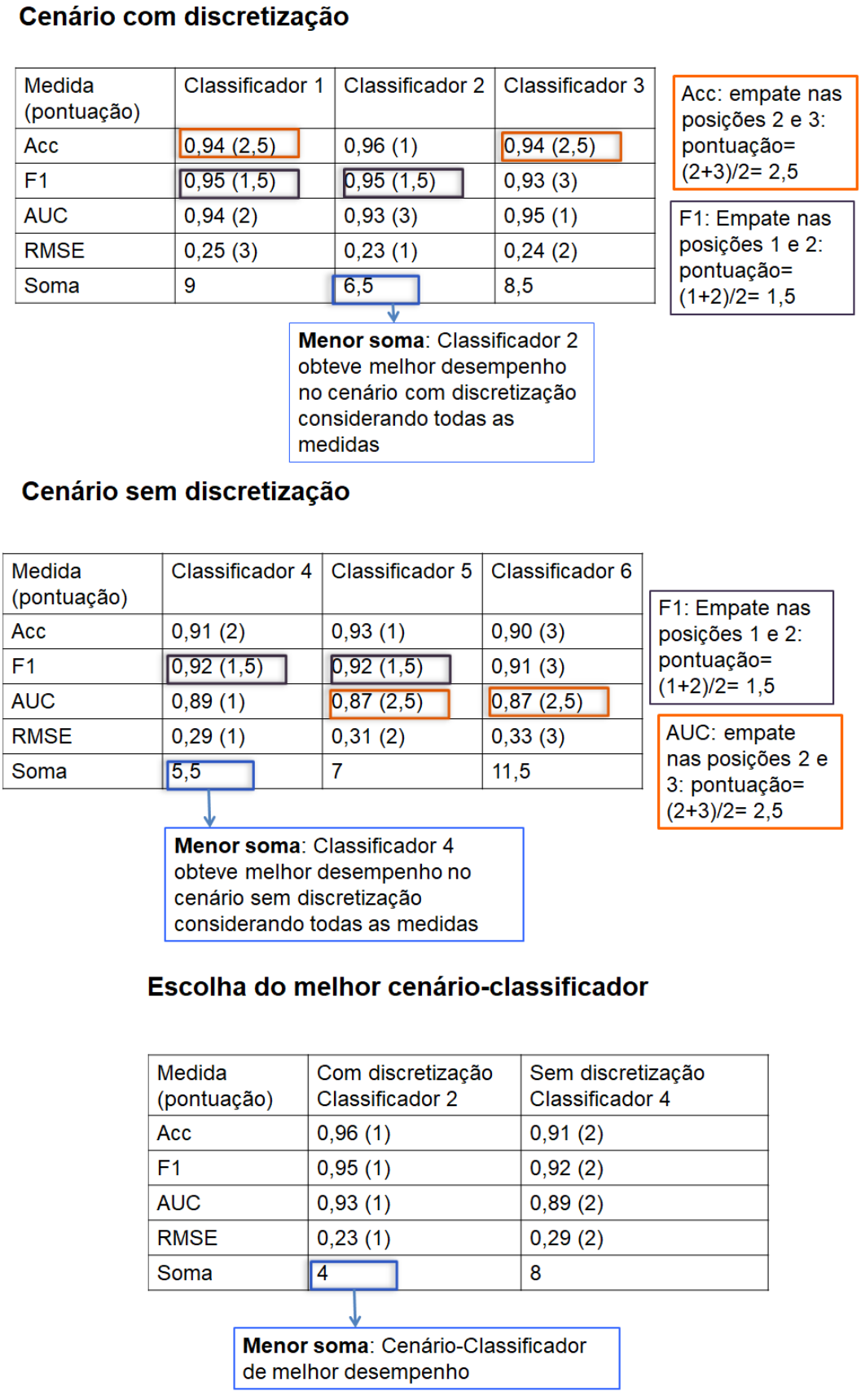


Figura 13: Exemplo de ranqueamento dos classificadores para a escolha do melhor cenário/classificador.

Formalmente, a escolha do melhor cenário de pré-processamento e classificador utiliza os algoritmos ilustrados na Figura 14.

No processo da Figura 12, cada saída da avaliação dos classificadores referente a cada cenário avaliado, consiste em uma estrutura de dados do tipo *HashMap* com r classificadores chamada de *lista_classificadores_medidas* em que cada classificador está associado aos valores obtidos por ele para as medidas de desempenho. Uma *lista_classificadores_medidas* para cada cenário é a entrada do Algoritmo 1 da Figura 14, que, por sua vez, tem como saída o melhor cenário/classificador. A estrutura *lista_classificadores_medida* tem a forma {‘Med_1’:{cl_1: v_1Med_1, cl_2: v_2Med_1,..., cl_r: v_rMed_1},‘ Med_2’:{cl_1: v_1Med_2, cl_2: v_2Med_2,..., cl_r: v_rMed_2}, ..., ‘Med_s’:{cl_1: v_1Med_s, cl_2: v_2Med_s ,..., cl_r: v_rMed_s}}, em que r denota o número de classificadores avaliados, s denota o número de medidas de desempenho adotadas, cl_i denota uma *String* contendo o nome do i-ésimo classificador e seus hiperparâmetros e v_iMed_k, denota o valor de desempenho obtido pelo i-ésimo classificador considerando a k-ésima medida de desempenho Med_k.

Para cada cenário, o Algoritmo 1 chama o Algoritmo 2 também ilustrado na Figura 14, passando *lista_classificadores_medidas* para obter o melhor classificador para cada cenário e preparar uma *lista_classificadores_medidas_modificada* com a forma {‘ Med_1’:{cen_1-$bestcl_1$: $vMed_1$_cen_1-$bestcl_1$, cen_2-$bestcl_2$: $vMed_1$_cen_2-$bestcl_2$,..., cen_t-$bestcl_t$: $vMed_1$_cen_t-$bestcl_t$}, ‘Med_2’:{cen_1-$bestcl_1$: $vMed_2$_cen_1-$bestcl_1$, cen_2-$bestcl_2$: $vMed_2$_cen_2-$bestcl_2$,..., cen_t-$bestcl_t$: $vMed_2$_cen_t-$bestcl_t$}, ..., ‘Med_s’:{cen_1-$bestcl_1$: $vMed_s$ _cen_1-$bestcl_1$, cen_2-$bestcl_2$: $vMed_s$ _cen_2-$bestcl_2$,..., cen_t-$bestcl_t$: $vMed_s$_cen_t-$bestcl_t$}} em que t denota o número de cenários, cen_j denota uma *String* contendo o nome do j-ésimo cenário, $bestcl_j$ denota uma String contendo o nome do melhor classificador e seus hiperparâmetros no cenário j, $vMed_k$_cen_j-$bestcl_j$, denota o valor de Med_k obtido pelo melhor classificador do j-ésimo cenário. A seguir, o Algoritmo 1 chama o Algoritmo 2 passando *lista_classificadores_medidas_modificada* para obter o melhor classificador e cenário.

As pontuações na criação de *list_rank* no Algoritmo 2 baseiam-se na posição de desempenho em *list* que está ordenada do melhor desempenho para o pior desempenho. Denotando med_{pos} como o valor da medida de desempenho na posição *pos* de *list*, para as medidas em que valores maiores representam melhor desempenho, med_{pos} será sempre maior ou igual à $med_{pos+1.}$ A lista *list_rank* é construída percorrendo-se *list* a partir de *pos*=0 e comparando-se valores adjacentes. No caso em que $med_{pos} > med_{pos+1}$, *list_rank[pos]* receberá *pos*, caso contrário (caso de empate), *list_rank[pos]* até o fim das posições empatadas

receberá a soma das posições consecutivas em que ocorrem empates dividida pelo número de posições empatadas.

Para as medidas em que valores menores representam melhor desempenho, *list_rank* também é construída percorrendo-se *list* a partir de *pos*=0 e comparando-se valores adjacentes, porém med_{pos} será sempre menor ou igual à med_{pos+1}. No caso em que $med_{pos}<med_{pos+1}$, *list_rank[pos]* receberá *pos*, caso contrário (caso de empate), *list_rank[pos]* até o fim das posições empatadas receberá a soma das posições consecutivas em que ocorrem empates dividida pelo número de posições empatadas.

No retorno do Algoritmo 2, o classificador (ou cenário-classificador) escolhido será o de menor pontuação total considerando todas as medidas de desempenho, pois as pontuações de cada medida baseiam-se na posição do desempenho em relação aos demais classificadores (ou cenários-classificadores) ordenado do melhor para o pior desempenho. Caso haja empate na primeira colocação no retorno do Algoritmo 2, o algoritmo possui uma ordem de prioridade pré-estabelecida. Para a escolha do melhor classificador dentro de cada cenário, tal ordem de prioridade vai do classificador mais interpretável para o menos interpretável, isto é, partindo dos Bayesianos, passando pelos baseados em árvore, pelos baseados em similaridade e, por fim, os baseados em funções matemáticas (ver ordem definida pelas Seções 3.2 a 3.9). Para a escolha do melhor cenário-classificador, prioriza-se aplicar menos pré-processamentos e, se possível, não criar instâncias artificiais. Sendo assim, a prioridade para bases balanceadas segue a seguinte ordem de cenários: (1) sem discretização e (2) com discretização. Para bases desbalanceadas, a prioridade segue a seguinte ordem de cenários: (1) sem discretização-sem sobreamostragem, (2) com discretização-sem sobreamostragem, (3) sem discretização-com sobreamostragem, (4) com discretização-com sobreamostragem.

Algoritmo 1: escolher_melhor_classificador_cenario

Entrada: uma *lista_classificadores_medidas* para cada cenário

Saída: melhor cenário e classificador

Método:

Criar um *HashMap* vazio chamado *lista_classificadores_medidas_modificada*;

Para cada cenario:

melhor_classificador=ranquear(*lista_classificadores_medidas*);

atualizar *lista_classificadores_medidas_modificada* (cenario, melhor_classificador) //para cada //medida, associar o valor de desempenho obtido pelo melhor_classificador no cenário

Fim-Para

retornar ranquear (*lista_classificadores_medidas_modificada*);

Algoritmo 2: ranquear

Entrada: *lista_classificadores_medidas*

Saída*:* melhor classificador (ou melhor cenario-classificador)

Método:

Criar um *HashMap* vazio chamado *medMap* ;

Para cada medida:

Se, para a medida, valores maiores representam melhor desempenho:

Criar uma lista de tamanho r chamada *list* contendo os valores da medida em ordem decrescente;

Senão:

Criar uma lista de tamanho r chamada *list* contendo os valores da medida em ordem crescente;

Criar uma lista de tamanho r chamada *list_rank* contendo as pontuações relativas às posições de desempenho usando list;

Criar uma lista de tamanho r chamada *list_classif* contendo cl_is correspondentes aos índices de *list*;

Criar um *HashMap* vazio chamado *classifMap*;

Para cada cl_i em *list_classif*:

Criar uma entrada em *classifMap* contendo {cl_i: list_rank[i]};// associa a cada //classificador a pontuação relativa à medida

Fim-Para

Criar uma entrada em *medMap* contendo {medida:classifMap};

Fim-Para

Percorrer *medMap* para criar um *HashMap classifMapTotal* com entradas {cl_i:totPont};

//cl_i: classificador (ou cenario-classif) e totPont: soma das pontuações do classif (ou do cenário-classif)

ordenar *classifMapTotal* em ordem crescente de pontuação total;

Se houver empate na primeira posição:

retornar cl_i do empatado de maior prioridade em *classifMapTotal*;

Senão: retornar o primeiro cl_i em *classifMapTotal*;

Figura 14: Algoritmos para ranqueamento e escolha do melhor cenário de pré-processamento e classificador.

4.2. SUGESTÕES DIAGNÓSTICAS, INDICAÇÕES SOBRE DADOS DE SAÚDE MAIS RELEVANTES E RECOMENDAÇÕES DE EXAMES FUTUROS

Suponha que um médico aplicou um conjunto de TNPs a um paciente *t* e coletou dados sobre fatores predisponentes incluindo idade, gênero e escolaridade (em anos de estudo) e informou todas as pontuações obtidas nos TNPs juntamente com os dados sobre os fatores predisponentes. A probabilidade deste paciente *t* possuir Demência, isto é, $P(Y_t = 1|\mathbf{x_t})$ será computada pelo classificador atual armazenado para diagnóstico de Demência, em que $\mathbf{x_t}$ representa o vetor de valores informados para os TNPs e fatores predisponentes. Ressalte-se que, se este classificador for um classificador probabilístico, tal probabilidade é inerentemente determinada pelo classificador. Contudo, se o classificador não for probabilístico, esta probabilidade será determinada por aproximação. Por exemplo, para o k-NN, a probabilidade será calculada usando a proporção dos vizinhos mais próximos que possuam a classe positiva a eles atribuída na base de treinamento. Cada classificador não probabilístico determinará, de forma aproximada, esta probabilidade de uma forma específica que dependerá do algoritmo de classificação.

Se $P(Y_t = 1|\mathbf{x_t}) > 0{,}6$, será sugerido um diagnóstico positivo para Demência e o Fator de Certeza (FC) será dado por $P(Y_t = 1|\mathbf{x_t})$. Por outro lado, se $P(Y_t = 1|\mathbf{x_t}) < 0{,}4$, será sugerido um diagnóstico negativo para Demência e o FC será dado por $1 - P(Y_t = 1|\mathbf{x_t})$. Se $0{,}4 \leq P(Y_t = 1|\mathbf{x_t}) \leq 0{,}6$, isto é, $(0{,}5 \leq FC \leq 0{,}6)$, nenhum diagnóstico será sugerido, pois os dados informados para aquele paciente são considerados insuficientes.

4.2.1. DADOS DE SAÚDE MAIS RELEVANTES PARA UM DIAGNÓSTICO SUGERIDO

Para indicar quais dados de saúde informados foram mais relevantes para o diagnóstico sugerido, se o cenário escolhido pelo processo de aprendizado automático foi com discretização, cada valor informado para cada exame/fator predisponente é atribuído individualmente e o FC é calculado. Se o FC resultante for menor que 0,6, o exame ou fator predisponente correspondente é considerado irrelevante para o diagnóstico sugerido. Então, os exames/fatores predisponentes considerados relevantes são posicionados em ordem decrescente de FC e os três primeiros (com maiores valores para o FC) são considerados os mais relevantes.

Todos os valores mostrados nesta seção e na Seção 4.2.2 são hipotéticos para didaticamente explicar como o modelo determina os exames mais relevantes para um diagnóstico sugerido ou recomenda exames futuros a serem aplicados aos pacientes.

Por exemplo, suponha que os dados informados para um paciente foram: idade = 74, gênero = "feminino", escolaridade = 16, MMSE = 23, CDT = 2, VFT = 10, depressão = "presença", CDR=?, Pfeffer=?, TMT - Parte A=?, Lawton=?, IQCODE=?, Stroop=?, Berg=?. O símbolo "?" denota dado não informado. Uma descrição detalhada sobre cada atributo será dada no Capítulo 6. Também, suponha, para cada atributo, os seguintes valores ou intervalos (obtidos por discretização supervisionada) possíveis:

- idade: 0-72, ≥73;
- gênero: "feminino", "masculino";
- escolaridade: 0-13, ≥14;
- MMSE: 0-17, 18-26, 27-30;
- CDT: 0, 1, 2, 3, 4, 5;
- VFT: 0-4, 5-11, ≥12;
- depressão: "ausência", "presença";
- CDR: 0, 0.5,1, 2, 3;
- Pfeffer: 0, 1-2, 3-30;
- TMT - Part A: 0-16, 17-59, ≥60;
- Lawton: 9, 10-27;
- IQCODE: 1-3.55, 3.56-5;
- Stroop: 0-15, ≥16;
- Berg: 0-51, 52-56.

Suponha que a saída do modelo seja um diagnóstico positivo para Demência com FC= 70% (considerado baixo). Para determinar quais são os dados de saúde mais relevantes que levaram a tal diagnóstico, primeiramente, FC_{idade} será computado, atribuindo à idade o segundo intervalo, já que idade=74, com todos os outros atributos como não informados:

idade ≥73,?,?,?,?,?,?,?,?,?,?,?,?,? → classificador→ $P(Y_t = 1|\text{idade} \geq 73) = 0{,}61$.

Portanto, para idade ≥73, $FC_{idade} = 0{,}61$.

Posteriormente, o mesmo é feito para os outros atributos informados:

- ?,gênero="feminino",?,?,?,?,?,?,?,?,?,?,?,? → classificador → $P(Y_t = 1|\text{gênero} = \text{"feminino"}) = 0{,}55$.

Como gênero="feminino" resultou em $FC_{gênero}$= 0,55, que é menor ou igual a 0,6, gênero não é considerado relevante para o diagnóstico sugerido.

- ?,?,escolaridade $\geq$ 14,?,?,?,?,?,?,?,?,?,?,? $\rightarrow$ classificador $\rightarrow$ $P(Y_t = 1|\text{escolaridade} \geq 14) = 0{,}4$.

  Como escolaridade$\geq$14 resultou em $FC_{escolaridade}$= 0,4 ($\leq$0.6), a escolaridade não é considerada relevante para o diagnóstico sugerido.
- ?,?,?,MMSE=18-26,?,?,?,?,?,?,?,?,?,? $\rightarrow$ classificador $\rightarrow$ $P(Y_t = 1|18 \leq \text{MMSE} \leq 26) = 0{,}81$.

  Então, FC_{MMSE}= 0,81.
- ?,?,?,?,CDT=2,?,?,?,?,?,?,?,?,? $\rightarrow$ classificador $\rightarrow$ $P(Y_t = 1|\text{CDT} = 2) = 0{,}78$.

  Então, FC_{CDT}= 0,78.
- ?,?,?,?,?,VFT=5-11,?,?,?,?,?,?,?,? $\rightarrow$ classificador $\rightarrow$ $P(Y_t = 1|5 \leq \text{VFT} \leq 11) = 0{,}63$.

  Então, FC_{VFT}= 0,63.
- ?,?,?,?,?,?,depressão="presença",?,?,?,?,?,?,? $\rightarrow$ classificador $\rightarrow$ $P(Y_t = 1|\text{depressão} = \text{"presença"}) = 0{,}75$.

  Então, $FC_{depressão}$=0,75.

Como FC_{MMSE}, FC_{CDT} e $FC_{depressão}$ obtiveram os três maiores valores, MMSE, CDT e depressão são indicados como os dados mais relevantes que levaram ao diagnóstico positivo para Demência.

Entretanto, se o cenário escolhido pelo processo automático de aprendizado foi sem discretização, a maneira previamente explicada para determinar os dados mais relevantes não pode ser aplicada se existem atributos que são contínuos. Neste caso, o ganho de informação será utilizado para escolher os dados mais relevantes para o diagnóstico sugerido. Usando o arquivo correspondente aos dados de treinamento de acordo com o cenário escolhido, os atributos serão ranqueados em ordem decrescente de valor de ganho de informação e os três atributos informados com maiores valores de ganho de informação serão indicados por serem os mais relevantes.

4.2.2. RECOMENDAÇÃO DE EXAMES FUTUROS A SEREM APLICADOS

Se um diagnóstico sugerido possui um FC baixo, TNPs são recomendados para serem aplicados ao paciente. O objetivo é ajudar a aumentar o FC através da indicação ao médico dos exames mais importantes que ainda não foram aplicados ou informados. Fatores

predisponentes são sempre informados porque são obrigatórios. Os exames recomendados são determinados através do cálculo das médias dos FCs para cada atributo. Cada intervalo ou valor possível, que um atributo pode assumir, produz um FC correspondente. Para cada atributo, a média dos FCs produzidos por seus intervalos/valores é calculada e, então, os atributos são ordenados em ordem decrescente de FC médio e os três atributos com maior valor de FC médio são recomendados.

Suponha, para o mesmo exemplo da seção anterior, que seja necessário indicar os exames ainda não informados que são importantes para aumentar o FC para o diagnóstico sugerido. No exemplo, são possíveis os seguintes valores ou intervalos (obtidos por discretização supervisionada) para cada atributo ainda não informado:

- CDR: 0, 0.5,1, 2, 3;
- Pfeffer: 0, 1-2, 3-30;
- TMT - Part A: 0-16, 17-59, ≥60;
- Lawton: 9, 10-27;
- IQCODE: 1-3.55, 3.56-5;
- Stroop: 0-15, ≥16;
- Berg: 0-51, 52-56.

Considere, como antes, os seguintes valores informados para o paciente: idade = 74, gênero = "feminino", escolaridade = 16, MMSE = 23, CDT = 2, VFT = 10, depressão = "presença". Para o CDR, não informado, o primeiro valor (0) é atribuído a ele, com todos os outros atributos não informados e FC_{CDR_0} é determinado:

idade ≥73, gênero="feminino", escolaridade≥14, MMSE= 18-26, CDT=2, VFT=5-11, depressão = "presença", CDR=0,?,?,?,?,?,? → classificador → $P(Y_t = 1|\mathbf{x_{cdr=0}}) = 0{,}1$.

Em que $\mathbf{x_{cdr=0}}$ denota o vetor contendo todos os dados informados para o paciente além do valor 0 para CDR.

Se $P(Y_t = 1|\mathbf{x_{cdr=0}})$ fosse maior que 0,5, FC_{CDR_0} receberia $P(Y_t = 1|\mathbf{x_{cdr=0}})$, porém, como $P(Y_t = 1|\mathbf{x_{cdr=0}}) \leq 0{,}5$, $FC_{CDR\_0} = 1 - P(Y_t = 1|\mathbf{x_{cdr=0}})$, resultando em $FC_{CDR\_0} = 0{,}9$.

O mesmo é feito para os outros valores que CDR pode assumir. Suponha que os resultados obtidos foram:

- $FC_{CDR\_0} = 1 - P(Y_t = 1|\mathbf{x_{cdr=0}}) = 1 - 0{,}1 = 0{,}9$;
- $FC_{CDR\_0.5} = 1 - P(Y_t = 1|\mathbf{x_{cdr=0.5}}) = 1 - 0{,}15 = 0{,}85$;
- $FC_{CDR\_1} = P(Y_t = 1|\mathbf{x_{cdr=1}}) = 0{,}81$;
- $FC_{CDR\_2} = P(Y_t = 1|\mathbf{x_{cdr=2}}) = 0{,}9$;

- $FC_{CDR\_3} = P(Y_t = 1|\mathbf{x_{cdr=3}}) = 0{,}95$.

Sendo assim, para o CDR a média dos FCs, denotada como FC_médio_CDR, será dada por: (0,9 + 0,85 + 0,81 + 0,9 + 0,95)/5, o que resulta em FC_médio_CDR =0,88.

O mesmo é feito para Pfeffer, TMT - Parte A, Lawton, IQCODE, Stroop e Berg. Para cada atributo, se FC_médio$\leq$ 0,6, o atributo é considerado irrelevante. Os três atributos com maiores valores para o FC_médio serão sugeridos como futuros exames a serem realizados.

Contudo, se o cenário escolhido pelo processo automático de aprendizado foi sem discretização, a maneira previamente explicada para recomendar exames futuros não poderá ser aplicada se existirem atributos contínuos. Neste caso, o ganho de informação será utilizado para recomendar exames futuros. Usando o arquivo correspondente aos dados de treinamento, os atributos serão ranqueados em ordem decrescente de valor de ganho de informação e os três atributos ainda não informados com maiores valores de ganho de informação serão recomendados por serem os mais relevantes.

Como mencionado anteriormente, o processo de aprendizado automático é executado periodicamente para cada doença (ou transtorno) de interesse. Com isso, o modelo de decisão proposto torna-se dinâmico por incluir os novos casos clínicos disponíveis com diagnósticos informados.

O próximo capítulo apresenta o sistema de apoio à decisão proposto, que implementa o modelo de decisão dinâmico descrito neste capítulo.

CAPÍTULO 5 – SISTEMA PARA APOIO AO DIAGNÓSTICO DE DEMÊNCIA, DA E CCL

Este capítulo descreve o CDSS proposto para auxiliar no diagnóstico de Demência, DA e CCL, o qual foi projetado para implementar o modelo de decisão dinâmico apresentado no Capítulo 4 e utilizar arquitetura aberta com componentes reutilizáveis.

5.1 PROCESSO DIAGNÓSTICO DE DEMÊNCIA, DA E CCL

O CDSS proposto visa dar suporte a um processo diagnóstico específico, seguindo diretrizes clínicas para diagnosticar Demência, DA e CCL. Estas diretrizes foram apontadas por médicos especialistas para especificação do processo diagnóstico, que está ilustrado na Figura 15. Neste processo diagnóstico, em um primeiro momento, o médico levanta o histórico do paciente e realiza testes para triagem de Demência. Se o médico suspeita que o paciente possa ter Demência, isto é, se o resultado da triagem for positivo para Demência, TNPs são realizados para confirmar o diagnóstico. Se o diagnóstico para Demência for confirmado como positivo, TNPs adicionais são realizados para diagnosticar se a Demência é do tipo DA ou não. Contudo, se o diagnóstico para Demência for confirmado como negativo, TNPs adicionais são realizados para diagnosticar se o paciente possui CCL ou não.

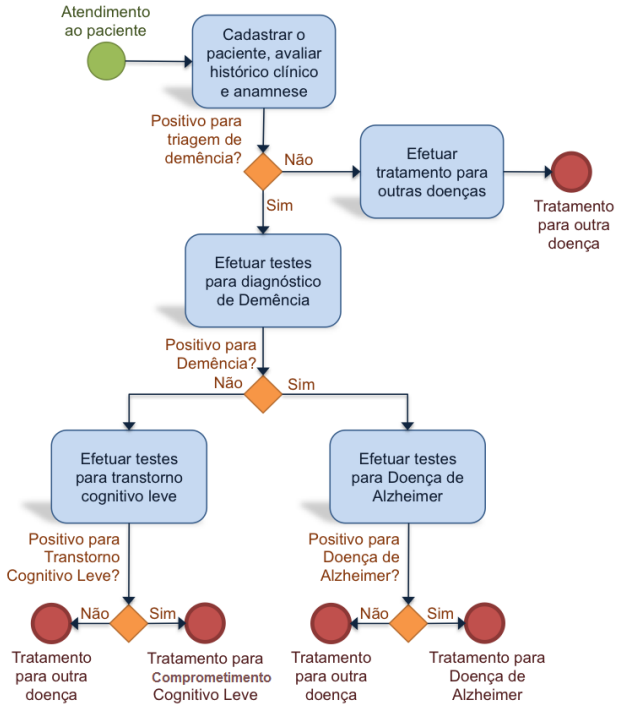


Figura 15: Processo diagnóstico para Demência, DA e CCL. Fonte: SEIXAS et al. (2014).

O CDSS desenvolvido auxilia os médicos a seguirem este processo diagnóstico, sendo que seu modelo de decisão é dinâmico e composto por um classificador binário para cada doença (ou transtorno) para prover o suporte necessário para a tomada de decisão quanto ao diagnóstico da respectiva doença (ou transtorno) que deve ser realizado pelo médico no respectivo instante. Sendo assim, esta implementação do modelo de decisão dinâmico genérico do Capítulo 4 é composta por um classificador binário para diagnóstico de Demência, que distingue os pacientes com Demência dos pacientes sem Demência, um classificador binário para DA que distingue pacientes com Demência devida a DA de pacientes com outros tipos de Demência e um classificador binário para CCL que distingue pacientes com CCL de pacientes normais (sem Demência de qualquer tipo ou CCL).

5.2 PRINCIPAIS COMPONENTES

A Figura 16 ilustra os principais componentes do sistema proposto. Um destes componentes é o modelo de decisão, que armazena os classificadores obtidos pelo processo automático de AMS que utiliza dados dos pacientes (casos clínicos) com diagnóstico confirmado contidos no banco de dados do sistema. Como dito anteriormente, o modelo de decisão contém três classificadores binários, um para cada doença/transtorno a diagnosticar: Demência, DA e CCL. Na implementação realizada, o processo de AMS é executado uma vez por semana e verifica, para cada doença/transtorno, se novos registros clínicos associados com diagnóstico foram inseridos no sistema desde a última reconstrução do modelo de decisão e, se foram inseridos, tal processo automaticamente retreina o modelo de decisão como detalhado na Seção 4.1 para a doença/transtorno em questão.

Outro componente do sistema é a interface de comunicação que: (1) recebe dos médicos requisições por suporte diagnóstico e registros clínicos contendo dados quanto a fatores predisponentes e resultados de exames (dados de saúde) para certo paciente sendo avaliado; e (2) retransmite os dados de saúde do paciente para o mecanismo de inferência e recomendação. O mecanismo de inferência e recomendação casa estes dados com os atributos do classificador correspondente no modelo de decisão e realiza a inferência (predição). Os resultados são o diagnóstico mais provável, um Fator de Certeza (FC) e os principais dados de saúde que conduziram a tal diagnóstico. Tais resultados são transmitidos para um dispositivo móvel através da interface de comunicação e são mostrados ao médico.

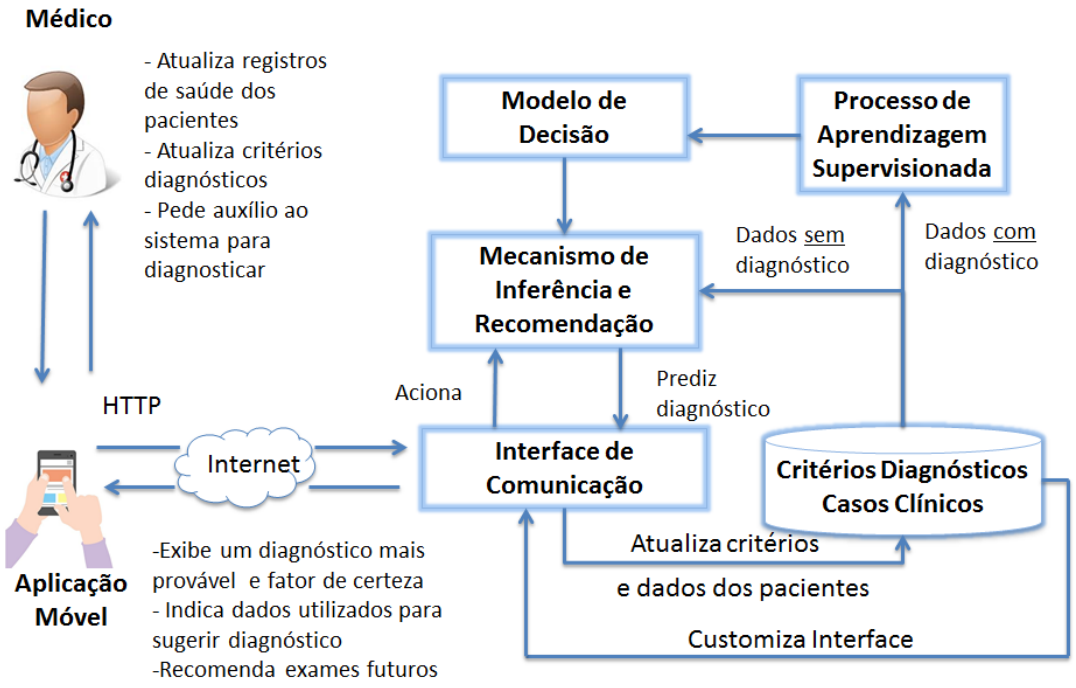


Figura 16: Principais componentes do CDSS.

5.3 APOIO AO DIAGNÓSTICO

Os resultados do apoio ao diagnóstico exibidos pelo CDSS seguem o processo ilustrado na Figura 17. Baseando-se nos registros de saúde previamente armazenados de certo paciente, o sistema avalia se é possível sugerir um diagnóstico para o paciente. O FC varia entre 50% e 100% e pode ser alto (maior ou igual a 80 %), baixo (entre 60% e 80%) ou insuficiente (menor ou igual a 60%). Por exemplo, se dados de saúde insuficientes foram informados, o FC obtido será menor que 60% e nenhuma sugestão diagnóstica será oferecida pelo sistema. Caso contrário, o CDSS sugere um diagnóstico positivo ou negativo para a doença (ou transtorno) sendo avaliada(o), um FC associado a este diagnóstico e os principais dados de saúde que levaram ao diagnóstico sugerido. Se o FC for alto, o próximo diagnóstico é avaliado seguindo o processo da Figura 15, caso contrário, o sistema sugere quais exames específicos devem ser aplicados ao paciente para aumentar o FC, desta forma, recomendando futuros exames objetivando confirmar ou refutar a hipótese diagnóstica inicial.

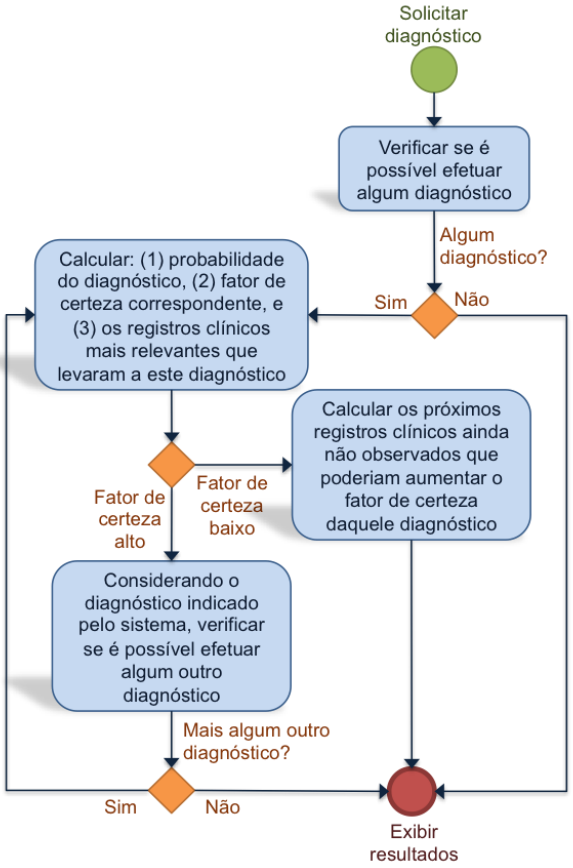


Figura 17: Fluxo de exibição dos resultados diagnósticos pelo CDSS.

5.4 REALIMENTAÇÃO DIAGNÓSTICA E CUSTOMIZAÇÃO DO SISTEMA

Os médicos podem concordar ou discordar do diagnóstico sugerido pelo sistema e, se discordarem, eles podem inserir o diagnóstico considerado correto. O diagnóstico considerado correto será levado em consideração quando o processo periódico de aprendizado for executado na próxima vez, pois comporá o conjunto de dados utilizado no retreinamento. Esta realimentação diagnóstica do especialista para a melhoria do modelo de decisão segue um paradigma de aprendizado denominado *active learning* (COHN, GHAHRAMANI E JORDAN, 1996; QUIONERO-CANDELA et al., 2009).

Além disso, os médicos podem customizar o CDSS com a inclusão de novos exames. Para isto, eles podem utilizar a interface do sistema para definir metadados relacionados a um novo exame, informando seu tipo (que pode ser contínuo ou categórico) e um intervalo válido (se o atributo for definido como contínuo) ou valores e rótulos válidos (se o atributo for definido como categórico). Os metadados são salvos no banco de dados do CDSS permitindo que o exame recém-definido possa ser escolhido e ter seu valor informado quando um paciente for avaliado. Posteriormente, o processo automático de aprendizado pode fazer a inclusão de tal exame no modelo de decisão como explicado na Seção 4.1.

5.5 INTERFACE DE USUÁRIO

As Figuras Figura 18Figura 20 ilustram as telas da interface com o usuário do CDSS proposto como visto de um navegador de um *smartphone*. Na Figura 18, à esquerda, a tela mostra dados de saúde simulados relacionados a um paciente fictício, e à direita, a tela ilustra opções disponíveis para a inclusão de um novo registro clínico para o paciente. Na Figura 19, à esquerda, a tela mostra a saída da inferência apresentada pelo CDSS, e à direita, a tela mostra as opções para a discordância do médico em relação aos resultados do diagnóstico apresentados pelo CDSS permitindo que ele indique o item de discordância e a justificativa para a discordância. Para o caso ilustrado, o paciente recebeu um diagnóstico positivo para Demência, com FC baixo (66,31%). Os registros clínicos mais relevantes que levaram ao diagnóstico foram: o resultado do CDR e o gênero do paciente. Como o FC foi baixo, o sistema também exibe quais itens clínicos (ainda não registrados) poderiam confirmar o diagnóstico: presença ou ausência de depressão e os resultados para os testes de Berg e Fluência Verbal.

A Figura 20 ilustra a tela para a inclusão de um novo tipo de exame: CERAD memória de lista de palavras, que está sendo definido como numérico (contínuo) com intervalo entre 0 e 30. Após salvo, tal exame será exibido na tela ilustrada pela Figura 18 (direita).

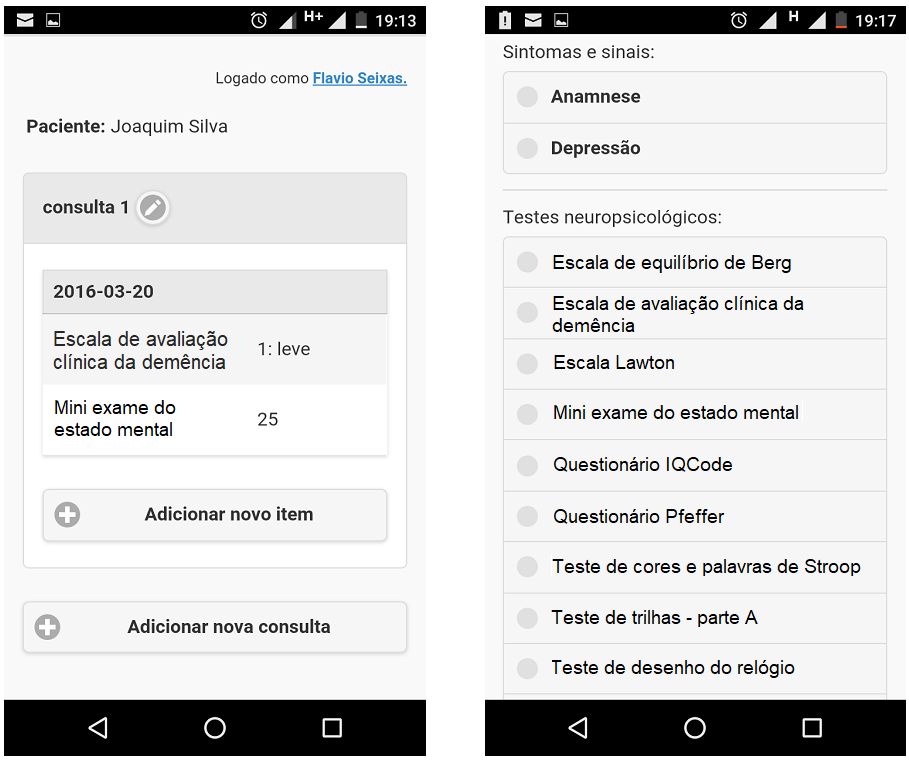


Figura 18: Registros de saúde do paciente (esquerda) e os TNPs que podem ser registrados no CDSS (direita).

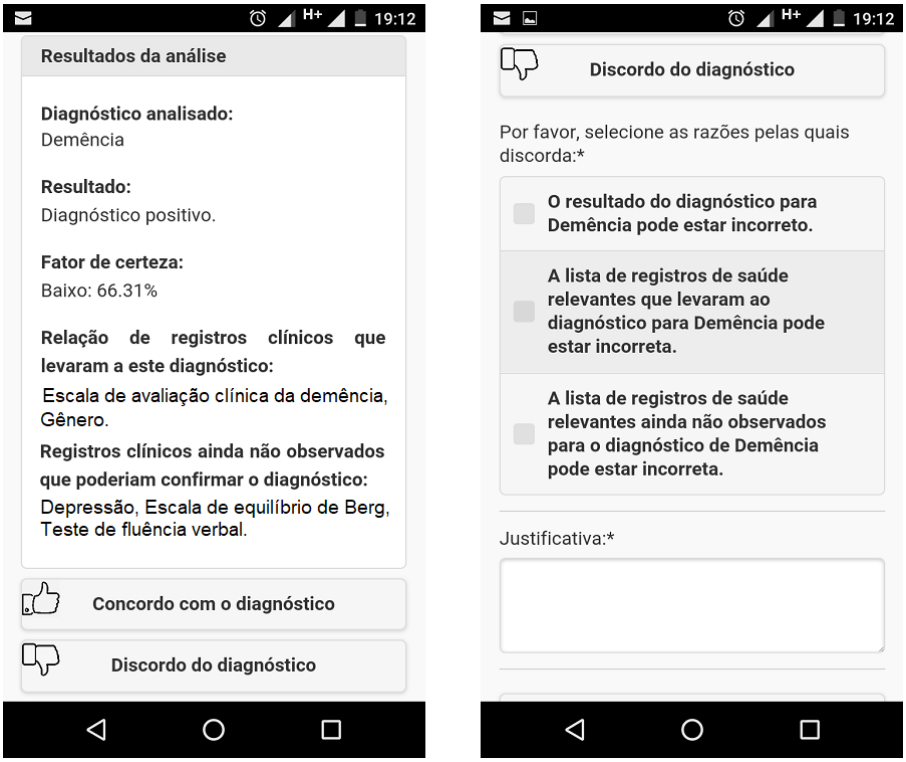


Figura 19: Sugestão de diagnóstico exibida pelo CDSS com base nos registros clínicos do paciente (esquerda) e a avaliação dos resultados pelo médico (direita).

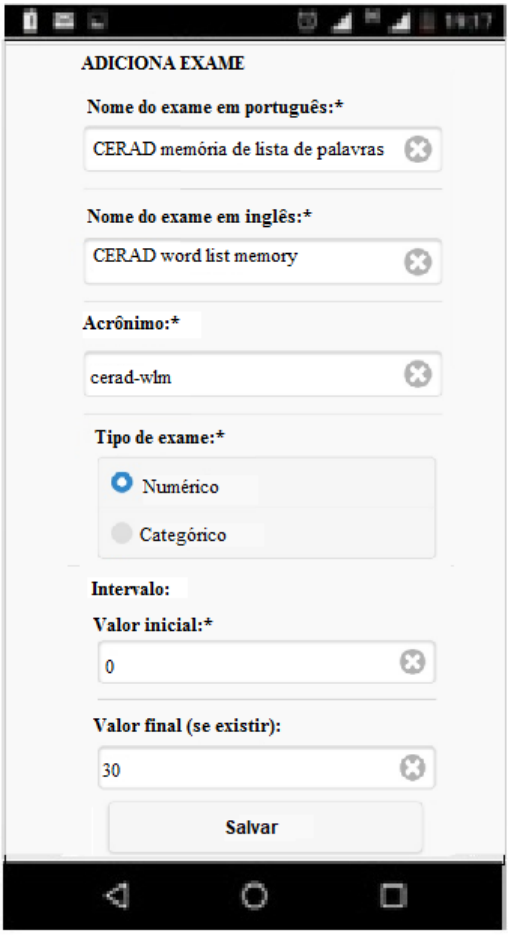


Figura 20: Exemplo de tela do CDSS para inclusão de exames.

5.6 PROJETO E ARQUITETURA

Durante a fase de projeto do CDSS, foram especificados e pesquisados componentes de um framework de aplicação web baseado na plataforma J2EE (*Java Enterprise Edition*) (KASSEM e TEAM, 2000). Esta plataforma usa uma arquitetura distribuída baseada em cliente-servidor, que oferece vantagens relacionadas a gerenciamento de recursos, componentes bem documentados e abertos e torna mais fácil prover requisitos de projeto importantes tais como escalabilidade, gerenciamento de falhas e qualidade de serviço.

De forma a reutilizar e desenvolver componentes e usar boas práticas de engenharia de software, como padrões de projeto e separação em camadas, foi escolhido *Java Server Faces* (JSF)[13] que inclui *Model-View-Controller* (MVC) como padrão arquitetural (LI e CUI, 2005). Objetivando prover componentes sofisticados da camada de apresentação que fossem adaptativos ao dispositivo cliente, seja ele móvel ou não, foi escolhido *PrimeFaces*[14]. Para prover a inferência, foram escolhidas duas APIs Java: a API GeNIe/SMILE e a API WEKA[15]. A API GeNIe/SMILE é invocada para inferência Bayesiana quando o classificador armazenado é uma RB discreta com três níveis como descrita por SEIXAS et al. (2014) e a API WEKA é invocada para a predição quando o classificador armazenado é de outro tipo.

Como framework de persistência foi escolhido *Hibernate*, que implementa a API de persistência Java (*Java Persistence* API - JPA) (KEITH e SCHINCARIOL, 2006). JPA é uma API Java padrão que descreve uma interface comum para frameworks de persistência de dados. Ela define meios para mapeamento entre objetos (*Entity Beans*) e bases de dados relacionais objetivando prover persistência em uma base de dados independente do Sistema Gerenciador de Banco de Dados (SGBD) escolhido. Como SGBD, foi escolhido o MySQL Server[16], um SGBD de código aberto muito popular. Para simplificar a construção da aplicação e os processos de *deployment*, é utilizado o MAVEN[17].

A Figura 21 ilustra os principais componentes que foram projetados ou integrados ao CDSS. A aplicação para dispositivos móveis (ou aplicativo) foi construída baseando-se nas especificações J2EE para aplicações web e em componentes *JavaBeans*. Esta arquitetura baseada em componentes implementa padrões importantes de projeto de software (YENER e THEEDOM, 2014), como injeção de dependências, padrão *facad*e e outros padrões de codificação.

[13] https://javaee.github.io/javaserverfaces-spec/
[14] http://www.primefaces.org/
[15] https://jar-download.com/artifacts/nz.ac.waikato.cms.weka/weka-stable/3.8.0/source-code
[16] https://dev.mysql.com/
[17] http://maven.apache.org/what-is-maven.html.

Páginas xHTML e todos os códigos Java estão hospedados em um servidor de aplicação. Portanto, as funcionalidades do aplicativo estão disponíveis através de um navegador web com suporte a AJAX e *JavaScript* (JS), sem a necessidade de instalação no lado cliente. Além disso, a interface gráfica de usuário permite deslizamento, toque e outras formas de interação típicas de usuários de dispositivos móveis. O navegador web troca mensagens HTTP através da Internet com um servidor de aplicação *Glassfish*[18], o qual provê o container *servlet* no qual o CDSS executa.

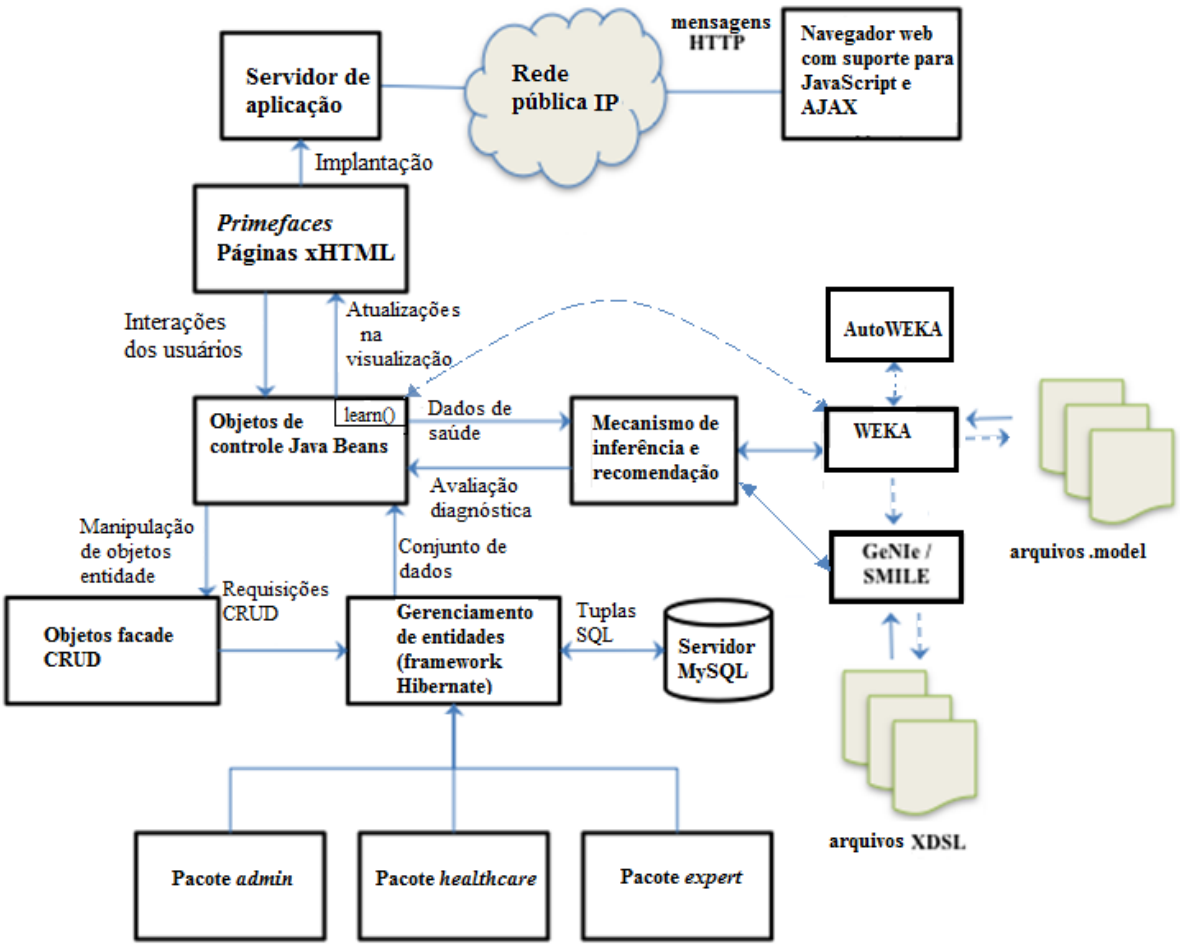


Figura 21: Arquitetura do CDSS proposto.

A camada de apresentação inclui *PrimeFaces*, uma biblioteca baseada *em Cascading Style Sheets* (CSS) e JS para projetos JSF que lida com componentes da interface de usuário. *PrimeFaces* provê suporte para lidar com eventos, processar interações do usuário, gerar páginas dinamicamente e gerenciar formulários. Este componente da camada de apresentação também é responsável pela validação dos formulários contendo os dados informados pelos usuários do sistema.

[18] http://www.oracle.com/technetwork/middleware/glassfish/overview/index.html

Requisições HTTP e lógica específica da aplicação são processadas por objetos do tipo *Controller*. Objetos *facade* contêm funções de criação, atualização e exclusão, sendo responsáveis por facilitar a visualização dos dados, buscas e atualizações.

Na arquitetura JPA, classes de entidade (*entity beans*) devem representar objetos do banco de dados. No CDSS proposto, as entidades de classe foram agrupadas em três pacotes: (1) classes de administração de sistema (pacote *admin*); (2) classes de gerenciamento de pacientes (pacote *healthcare*); e (3) classes relacionadas ao modelo de decisão (pacote *expert*). As classes de administração de sistema representam objetos de dados relacionados à administração dos usuários, como senhas, preferência e outros parâmetros de configuração. Os objetos das classes de gerenciamento de pacientes armazenam informação demográfica dos pacientes, dados relacionados às consultas e todo o histórico de registros de saúde dos pacientes. Portanto, tais classes representam *entity beans* que lidam com os dados dos pacientes, incluindo-se dados clínicos de saúde informados pelos médicos ao usarem o sistema (como as pontuações obtidas nos TNPs). Finalmente, as classes do pacote *expert* representam objetos que armazenam dados relacionados ao modelo de decisão e aos resultados do processo automático de aprendizado como os nomes dos arquivos contendo os últimos classificadores obtidos, o desempenho estimado para estes classificadores, o instante de início e a duração do processo de aprendizado.

O mecanismo de inferência e recomendação é responsável pela obtenção das sugestões diagnósticas, cálculo dos FCs e listagem das avaliações diagnósticas possíveis que podem ser realizadas com base no histórico de saúde do paciente. Ele também indica os dados de saúde mais relevantes que levaram ao diagnóstico sugerido e recomenda exames futuros no caso de FC baixo. Ele casa os dados de saúde com os atributos do classificador correspondente e realiza a inferência/predição invocando a API GeNIe/SMILE ou a API WEKA. Os classificadores treinados são armazenados em arquivos XDSL para as RBs contendo nós, estados e parâmetros aprendidos (CPDs) ou, para outros tipos de classificadores, arquivos com a extensão *model* usada pela ferramenta WEKA para armazenar os classificadores aprendidos. O resultado da predição é um diagnóstico (negativo ou positivo) e uma probabilidade associada usada para computar o FC.

Toda a vez que um médico informa um diagnóstico para certo paciente, os fatores predisponentes e os resultados dos exames para este paciente são armazenados na tabela *learning_data* do banco de dados juntamente com o diagnóstico provido e o instante em que o diagnóstico foi informado.

Um método programado, chamado *learn*, pertencente a um objeto do tipo *Controller* chamado *LearningController*, é disparado semanalmente durante um período de baixo uso do sistema (0:00 de domingo). Este método implementa o processo da Figura 12 verificando, para cada doença/transtorno, se existem novos registros clínicos com diagnóstico desde o último aprendizado realizado para a doença em questão. Quando existem, é necessário retreinar o modelo de decisão, de forma que o método seleciona os dados de saúde de todos os pacientes diagnosticados como positivo ou negativo para a doença/transtorno em questão e converte tais dados para o formato ARFF [19] para formar o conjunto inicial de dados de treinamento. Tal conversão é necessária, por que as próximas etapas são realizadas invocando-se a API WEKA, a API da ferramenta AutoWEKA e eventualmente a API GeNIe/SMILE. A ferramenta AutoWEKA (THORNTON et al., 2013) realiza a otimização dos hiperparâmetros possuindo a habilidade de buscar eficientemente no espaço de hiperparâmetros, para um dado algoritmo de classificação e duração máxima pré-definida, valores de combinações que resultem na minimização do percentual de erros na classificação. Na arquitetura do CDSS, a API AutoWEKA é utilizada para o ajuste automático dos hiperparâmetros dos classificadores avaliados no processo da Figura 12. A avaliação dos classificadores utiliza *leave-one-out* e as medidas de desempenho escolhidas para tal avaliação estão resumidas na Tabela 6 do Capítulo 3. As medidas Acc, F1 e AUC foram escolhidas por serem muito comuns em aplicações médicas. A medida RMSE é calculada usando a média dos erros probabilísticos ao quadrado das tuplas de teste (ver Expressão 50 no Capítulo 3). Para cada tupla a predizer, a saída do classificador possui duas partes: o rótulo da classe e a probabilidade de a tupla pertencer a esta classe. O RMSE leva em consideração este valor de probabilidade retornado pela predição. Para o sistema proposto, é desejável que o valor do RMSE seja mínimo, ou seja, que além de se classificar as tuplas corretamente, as probabilidades de pertencimento destas tuplas às respectivas classes se aproximem de 100%, elevando o Fator de Certeza (FC) que é exibido juntamente com o diagnóstico sugerido, o qual é levado em consideração pelo médico ao tomar sua própria decisão diagnóstica. Por isso, o RMSE também foi escolhido para avaliar os classificadores.

Na Figura 21, as interações disparadas exclusivamente pelo método *learn* estão ilustradas por linhas tracejadas. O controlador *LearningController* é um tipo especial de controlador que não interage com a camada de apresentação, apenas seleciona do banco de dados os dados usados no aprendizado, e, quando necessário, atualiza objetos do pacote

[19] https://www.cs.waikato.ac.nz/ml/weka/arff.html

expert armazenando no sistema de arquivos os arquivos com extensão XDSL ou *.model* gerados pelo processo de aprendizado. Embora k-NN seja um classificador do tipo *lazy* (ver Seção 3.8), caso ele seja escolhido, a API WEKA também o armazenará como um arquivo *.model* que codificará o conjunto de dados de treinamento correspondente juntamente com os hiperparâmetros escolhidos para o k-NN. O conjunto de dados utilizado no treinamento do modelo obtido também é armazenado no sistema de arquivos para permitir o ranqueamento doa atributos pelo valor do ganho de informação que poderá ser necessário como explicado na Seção 4.2.

O método *learn* é anotado com *@Schedule* e usa o serviço *Timer* que executa no *container* e registra o EJB correspondente (no caso, o *LearningController*) para os métodos de retorno de chamada. Tal serviço rastreia os temporizadores e respectivos agendamentos e também persiste os temporizadores em caso de falha ou desligamento do servidor. O método anotado é invocado pelo agendador do *container* nos tempos ou intervalos (se o método for periódico) especificados nos argumentos da anotação. O temporizador começa quando o EJB é implantado.

A Figura 22 ilustra o Diagrama Entidade-Relacionamento (DER) para o banco de dados do CDSS com foco nas tabelas que armazenam dados usados na predição ou aprendizado. Dados de saúde dos pacientes usados na predição são armazenados na tabela *patient_data* (que armazena os dados relacionados aos fatores predisponentes) e na tabela *health_data* (que armazena as pontuações obtidas na aplicação dos exames). Os exames são organizados por consultas (*reviews*). A predição também consulta a tabela *item_description* e as tabelas associadas *option_value* e *interval_value* que contêm metadados relacionados aos dados de saúde dos pacientes. Tais metadados são usados para casar os dados de saúde com os atributos do classificador correspondente. Para realizar a predição, a tabela *model* é consultada por diagnóstico (doença ou transtorno) e o modelo mais recente é obtido. A Figura 22 também ilustra a tabela *learning_data* que é consultada pelo método agendado *learn* para realizar o processo de aprendizado automático. A tabela *model* armazena os resultados do processo de aprendizado incluindo os nomes dos arquivos que armazenam os classificadores obtidos (campo *modelFileName*), desempenho estimado de tais classificadores, data/hora em que o processo começou (campo *creationDate*) e duração do processo de aprendizado em minutos (campo *duration*). Uma vez por semana, o método *learn* verifica se novos diagnósticos para cada doença/transtorno foram inseridos desde o último aprendizado realizado consultando a tabela *learning_data* pelos campos *creationDate* e *itemDescriptionFK* (cada doença/transtorno tem um registro *itemDescription* correspondente

na tabela *item_description*). Se existe pelo menos um registro com campo *creationDate* posterior ao campo *creationDate* do último modelo correspondente armazenado, um novo processo de aprendizado é acionado para a respectiva doença/transtorno. As tabelas *item_description*, *option_value* e *interval_value* são atualizadas quando um médico define um novo tipo de exame e suas modificações se refletem na interface do sistema.

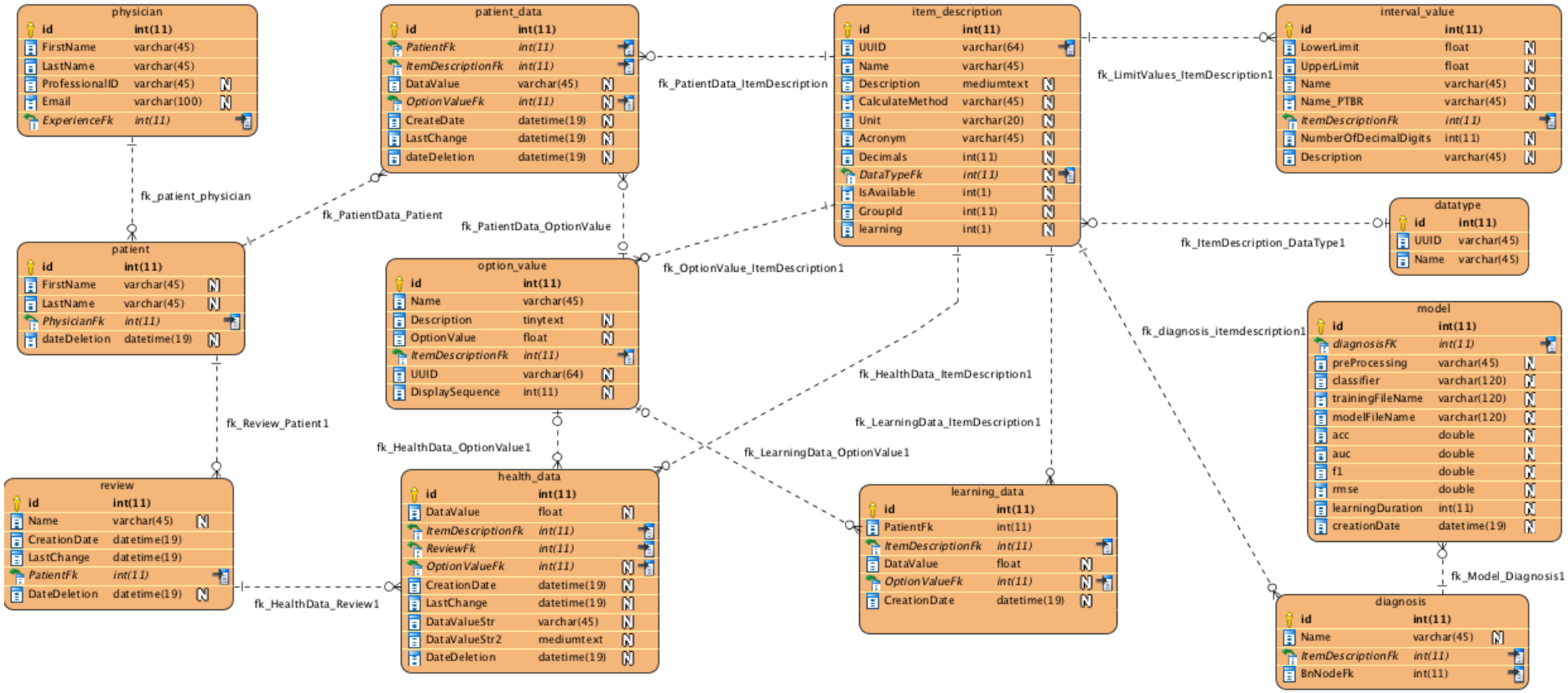


Figura 22: Diagrama Entidade-Relacionamento para o banco de dados do CDSS para as tabelas relacionadas à predição e ao aprendizado.

CAPÍTULO 6 – RESULTADOS EXPERIMENTAIS

Este capítulo detalha os conjuntos de dados utilizados na realização de testes experimentais, além de apresentar os resultados alcançados na aplicação do modelo de decisão dinâmico ao CDSS que auxilia no diagnóstico de Demência, DA e CCL. Também são apresentados os resultados de desempenho do modelo obtido considerando-se dados não utilizados no treinamento e a discussão de tais resultados. Além disso, este capítulo compara o conjunto de exames considerados mais relevantes de acordo com o modelo de decisão obtido com o conjunto de exames considerados mais relevantes por médicos especialistas.

6.1 DESCRIÇÃO GERAL DOS CONJUNTOS DE DADOS E EXAMES

Para a realização de testes experimentais, dados de pacientes de dois centros de saúde diferentes foram utilizados: do Centro de Doença de Alzheimer e outras Desordens Mentais na Velhice (CDA) do Instituto de Psiquiatria da Universidade Federal do Rio de Janeiro (UFRJ) e do Centro de Referência em Atenção à Saúde dos Idosos (CRASI) do Hospital Universitário Antônio Pedro da Universidade Federal Fluminense (UFF).

A Tabela 8 apresenta os exames mais frequentemente aplicados para diagnóstico de Demência, DA e CCL no CDA e no CRASI de acordo com os respectivos conjuntos de dados. Note que existem 6 exames em comum, mas existem 5 exames usados somente no CDA e 7 exames usados somente no CRASI. Os atributos idade, gênero e escolaridade (não mostrados na Tabela 8) são fatores predisponentes considerados por ambos os conjuntos de dados, CDA e CRASI. Atributos com valores que não são de fato contínuos ($\in$ R) foram definidos como tal nos casos em que o número de níveis seria maior que 7 se os atributos fossem considerados categóricos.

A seguir, será feita uma breve descrição dos exames presentes na Tabela 8.

O escore clínico CDR (MORRIS, 1993) tem 5 pontuações e visa avaliar os estágios da demência: 0 para pacientes sem alteração; 0.5 para pacientes com alteração questionável; 1 para pacientes com demência considerada leve; 2 para pacientes com demência considerada moderada e 3 para pacientes com demência considerada severa.

O teste conhecido por MMSE ou MEEM (Mini Exame do Estado Mental) (FOLSTEIN, FOLSTEIN e MCHUGH, 1975), é um TNP muito comum aplicado para a avaliação cognitiva sendo considerado um exame de rastreio. Sua escala varia de 0 a 30 onde pacientes normais tendem para a pontuação 30.

Tabela 8: Exames mais frequentes no CDA e no CRASI.

Exames	Domínio	Definido como contínuo	CDA	CRASI
Clinical Dementia Rating (CDR) (MORRIS,1993)	0, 0.5,1,2,3		X	X
Mini Mental State Exam (MMSE) (FOLSTEIN, FOLSTEIN e MCHUGH, 1975)	$N \in [0,30]$	X	X	X
Verbal Fluency Test (VFT) (PASSOS et al., 2011)	N	X	X	X
Clock Drawing Test (CDT) (SHULMAN, SHEDLETSKY e SILVER, 1986)	$N \in [0,5]$		X	X
Depressão (YESAVAGE e SHEIKH, 1986)	ausência, presença		X	X
Lawton (LAWTON e BRODY, 1969)	$N \in [9,27]$	X	X	X
Trial Making Test (TMT) Parte A[a] (TOMBAUGH, 2004)	N	X	X	
Teste de cores e palavras de Stroop (STROOP, 1992)	N	X	X	
Escala de equilíbrio de Berg (MIYAMOTO et al., 2004)	$N \in [0,56]$	X	X	
Questionário Pfeffer (PFEFFER et al., 1982)	$N \in [0,30]$	X	X	
Questionário IQCODE (JORM, 2004)	[1,5]	X	X	
Índice Katz (KATZ, 1983)	$N \in [0,6]$			X
TMT – Parte A -1-OK[b]	0,1			X
TCA (ZERBINI et al., 2009) percentual de respostas incorretas	$N \in [0:100]$	X		X
TCA percentual de respostas omitidas	$N \in [0:100]$	X		X
TCA tempo de reação	N	X		X
TCA variabilidade do tempo de reação	N	X		X
CERAD memória de lista de palavras (BERTOLUCCI et al., 2001)	$N \in [0:30]$	X		X

[a.] Tempo que o paciente leva para realizar o teste em segundos.

[b.] O resultado é igual a 1 caso o paciente consiga completar o teste adequadamente ou 0 caso contrário.

O teste VFT (PASSOS et al., 2011) é um TNP para avaliar a fluência verbal do paciente no qual ele tenta falar a quantidade máxima de palavras que conseguir lembrar para uma categoria semântica proposta em um tempo de 60 segundos. O teste detecta alterações (pontuações baixas) no caso de doenças neurodegenerativas ou transtornos mentais. A pontuação final é o número de palavras corretas que o paciente consegue lembrar e, portanto, a pontuação mais baixa é zero e não existe limite teórico superior.

O TNP conhecido por CDT (SHULMAN, SHEDLETSKY e SILVER, 1986) ou teste do relógio é uma ferramenta de avaliação cognitiva em que o paciente desenha um relógio e o médico avalia o resultado para atribuir uma pontuação que varia de 0 a 5 de acordo com a presença ou ausência de elementos coerentes e a qualidade do traço.

A depressão é uma comorbidade muito comum que pode aparecer associada à Demência. A escala GDS (YESAVAGE e SHEIKH, 1986) baseia-se em um questionário com perguntas direcionadas ao paciente sendo frequentemente utilizada para detectar a depressão.

Uma pontuação acima de 5 significa que o paciente está deprimido. Em ambos os conjuntos de dados, a depressão é representada como um atributo booleano indicando a presença ou ausência de depressão no paciente.

A escala Lawton (LAWTON e BRODY, 1969) é também conhecida por Avaliação das Atividades Instrumentais da Vida Diária (AIVD) sendo usada para avaliar a independência do paciente na realização de atividades cotidianas. Ela consiste em 9 perguntas e para cada uma delas é atribuída uma pontuação: 1 (para paciente dependente), 2 (para paciente parcialmente dependente) ou 3 (para paciente independente). Sendo assim, a pontuação final varia de 9 a 27 e quanto maior a pontuação, mais independente é o paciente na realização de suas atividades diárias.

O teste de trilhas (TMT) (TOMBAUGH, 2004) avalia a atenção seletiva, o processamento das percepções e a habilidade mental e consiste em ligar com um lápis 25 números em ordem crescente (TMT – Parte A) ou números alternados com letras (TMT- Parte B). No conjunto de dados do CDA, a duração em segundos que o paciente leva para realizar o teste TMT-Parte A é um atributo enquanto no conjunto de dados do CRASI o atributo para tal teste é Booleano: 1 se o paciente é capaz de executar tal teste adequadamente ou 0 caso contrário.

O teste Stroop (STROOP, 1992), também conhecido como teste das cores e palavras de Stroop (*Stroop color-word naming test*), é um TNP para medir as funções executivas e de atenção em que nomes de cores são mostrados ao paciente de forma colorida e ele deve indicar que cor enxerga. As palavras podem confundir o paciente porque palavras expressando certa cor podem ser exibidas em uma cor diferente, por exemplo, a palavra “amarelo” pode ser exibida em azul e um paciente que deveria indicar “azul” pode dizer incorretamente “amarelo”. A pontuação para o teste Stroop no CDA é derivada do tempo médio em milissegundos que um paciente leva para decidir cada palavra com limite inferior igual a 0 e sem limite superior teórico.

A escala de equilíbrio de Berg (MIYAMOTO et al., 2004) é um teste clínico para avaliar as habilidades de equilíbrio dinâmico e estático de um paciente sendo observado. A pontuação final varia entre 0 e 56 em que, quanto mais baixa a pontuação, mais prejudicado está o equilíbrio.

O questionário Pfeffer (PFEFFER et al., 1982) é usado para avaliar a independência do paciente na execução de certas tarefas. A pontuação final varia de 0 a 30 e pontuações mais altas expressam maior dependência.

O questionário IQCODE (JORM, 2004) é aplicado para detectar a presença de declínio cognitivo considerando-se a última década e baseando-se no relato de um acompanhante do paciente sendo avaliado. O questionário é composto, em sua versão original, de 26 itens, com questões organizadas em uma escala *likert* com cinco opções e pontuações correspondentes: (1) para muito melhor, (2) para um pouco melhor, (3) para sem mudança, (4) para um pouco pior, e (5) para muito pior. O resultado final é obtido pela soma das pontuações dividida pelo número de itens variando de 1 a 5 e assumindo valores decimais. Um resultado final igual a 3 significa que, em média, não houve alteração na capacidade cognitiva, maior que 3 significa que houve declínio na capacidade cognitiva e menor que 3 significa que houve uma melhoria na capacidade cognitiva.

O índice Katz de independência nas atividades da vida diária (KATZ, 1983), comumente conhecida por índice das Atividades Básicas da Vida Diária (AVD) de Katz, é usado para avaliar a habilidade do paciente em realizar as atividades diárias de forma independente. O índice avalia seis funções incluindo tomar banho, vestir-se, ir ao banheiro, locomover-se, alimentar-se e continência urinária e fecal. A cada paciente é atribuído sim ou não para independência em cada uma das seis funções. Uma pontuação de 6 indica independência completa, 5 indica dependência leve, 4 ou 3 indica dependência moderada e menor ou igual a 2 indica dependência severa.

O teste TCA (Teste Computadorizado de Atenção visual) (ZERBINI et al., 2009) tem duração de 1 minuto e meio. Vários estímulos visuais contendo formas distintas são mostrados ao paciente em sequência no centro de uma tela, cada estímulo com duração de 250 ms. O paciente deve clicar toda a vez que ele vê a forma de uma estrela. Quatro parâmetros são medidos: (1) o percentual de respostas incorretas (TCA-E), que mede a impulsividade; (2) o percentual de respostas omitidas (TCA-O), relacionado à desatenção; (3) o tempo de reação (TCA-TR) em milissegundos, relacionado à velocidade da reação motora ao estímulo visual; e (4) a variabilidade do tempo de reação (TCA-VAR), que reflete a concentração.

No teste CERAD memória de lista de palavras (BERTOLUCCI et al., 2001), 10 palavras não relacionadas entre si são lidas em voz alta, uma por uma, pelo paciente (ou médico, em caso de dificuldade de leitura por parte do paciente), na velocidade de uma palavra a cada 2 segundos. O paciente deve repetir as palavras lidas imediatamente após a última palavra ser lida por um período máximo de 90 segundos. O procedimento é repetido mais duas vezes. A pontuação final é obtida pela soma do número de palavras lembradas e repetidas nas três tentativas, sendo que a pontuação máxima que pode ser obtida é de 30 pontos.

6.2 TESTES INICIAIS UTILIZANDO DADOS DO CDA PARA CONSTRUIR O MODELO DE DECISÃO E DADOS DO CRASI PARA TESTAR A GENERALIZAÇÃO DO MODELO OBTIDO

Nos testes inicialmente realizados, dados do CDA foram utilizados para testar o processo automático que constrói o modelo de decisão escolhendo o melhor classificador e cenário de pré-processamento para cada doença/transtorno. Dados do CRASI, coletados posteriormente, foram utilizados para testar os classificadores obtidos pelo processo de aprendizado com dados novos não utilizados no treinamento. CARVALHO et al. (2018) descreveram resultados experimentais de um processo de aprendizado automático utilizando a base do CDA, porém, o processo descrito neste trabalho não ajustava automaticamente hiperparâmetros dos classificadores (utilizava hiperparâmetros em configurações padrão da WEKA) e a sobreamostragem era sempre aplicada no caso de bases desbalanceadas, isto é, para bases desbalanceadas, não era testado o cenário sem desbalanceamento.

Posteriormente, foram realizados testes finais utilizando os mesmos dados do CDA e os dados do CRASI utilizados nos testes iniciais para reconstruir o modelo de decisão dinâmico. Para a realização destes testes finais, dados adicionais do CRASI foram coletados para testar a generalização dos classificadores obtidos utilizando dados novos não usados no treinamento.

A presente seção detalha os dados utilizados e os resultados dos testes iniciais e a Seção 6.3 detalha os dados utilizados e os resultados dos testes finais.

6.2.1 CONJUNTOS DE DADOS UTILIZADOS NOS TESTES INICIAIS

A Figura 23 (a-c) mostra o número de pacientes com diagnóstico confirmado (positivo ou negativo) do CDA cujos dados foram usados para testar o processo automático de aprendizado e a Figura 23 (d-f) mostra o número de pacientes com diagnóstico confirmado do CRASI cujos dados foram usados para testar o modelo de decisão obtido formando conjuntos de teste em separado. De acordo com o processo diagnóstico ilustrado pela Figura 15, dados de pacientes diagnosticados como dementes podem compor os conjuntos de dados de DA enquanto dados de pacientes diagnosticados como não dementes podem compor os conjuntos de dados de CCL. Note-se que, dos 12 pacientes diagnosticados como positivos para Demência no CRASI (ver Figura 23 (d)), apenas 4 pacientes passaram por todo o processo diagnóstico sendo diagnosticados quanto à Demência ser do tipo DA ou não (ver Figura 23 (e)). Os outros 8 pacientes foram diagnosticados como positivos para Demência de todos os tipos, sem diagnóstico se a Demência é devida à DA ou não.

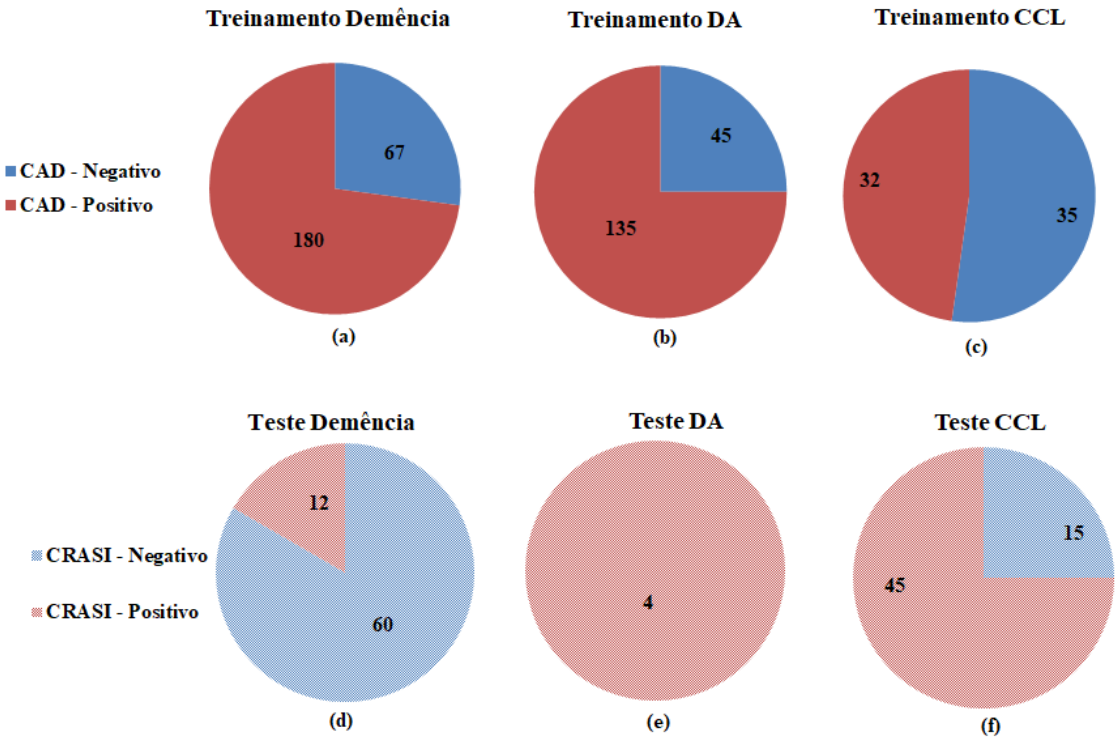


Figura 23: Número de pacientes diagnosticados nos conjuntos de dados para Demência, DA e CCL usados nos testes iniciais para treinamento (a-c) e teste (d-f).

6.2.2 RESULTADOS ALCANÇADOS PELO PROCESSO AUTOMÁTICO DE APRENDIZADO DO MODELO

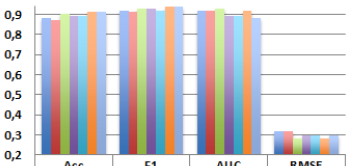
O método *learn* foi executado usando dados rotulados do CDA (ver Figura 23 (a-c)) e levou 182 minutos para ser executado em *background* por um processador Intel de 1.6 GHz Intel com memória de 4GB. A maior parte deste tempo (cerca de 93%) foi gasto pela otimização de hiperparâmetros realizada pela API AutoWEKA. Todos os resultados de desempenho estimado e ranqueamento dos classificadores foram registrados pelo processo, sendo apresentados na próxima seção.

6.2.2.1 DESEMPENHO ESTIMADO PARA OS CLASSIFICADORES E CENÁRIOS AVALIADOS E SELEÇÃO DO MELHOR CLASSIFICADOR E CENÁRIO PARA DEMÊNCIA, DA E CCL

A Figura 24 ilustra os desempenhos estimados para os classificadores avaliados pelo processo de aprendizado para o diagnóstico de Demência. RMSE é a única medida em que valores mais próximos a zero indicam melhor desempenho. Para as demais medidas, valores mais próximos a 1 indicam melhor desempenho. Para cada cenário de pré-processamento, hiperparâmetros específicos foram escolhidos pela AutoWEKA (ver Tabela 7). Como o

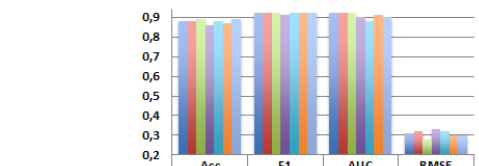
conjunto de dados de Demência estava desbalanceado (ver Figura 23 (a)), quatro cenários de pré-processamento foram avaliados, combinado a aplicação ou não da discretização e da sobreamostragem. As Tabelas 9-12 mostram os resultados do processo de ranqueamento que foi realizado para os quatro cenários. Neste processo de ranqueamento, para cada classificador/medida de desempenho é dada uma pontuação representando a posição do classificador em relação aos demais classificadores considerando a medida. Então, conforme detalhado na Seção 4.1.1, para cada classificador, uma pontuação total considerando todas as medidas (Acc, F1, AUC, RMSE) é obtida através da adição das pontuações para cada medida. A seguir, os classificadores são ordenados em ordem crescente de pontuação total e o primeiro classificador (com pontuação total mais baixa) é considerado o melhor. A Tabela 13 mostra o ranqueamento realizado para o melhor classificador de cada cenário e a escolha do melhor cenário de pré-processamento e classificador. Para Demência, o vencedor foi o cenário com discretização e sem sobreamostragem e o classificador RF com número de árvores igual a 230, e número de atributos aleatoriamente escolhidos igual a 4 e máxima profundidade das árvores igual a 5.

Desempenho para Demência com discretização e sem sobreamostragem



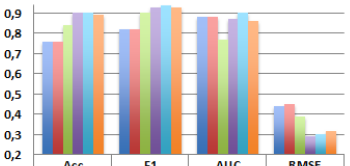
	Acc	F1	AUC	RMSE
BN	0,88	0,92	0,92	0,32
NB	0,87	0,91	0,92	0,32
A1DE	0,9	0,93	0,93	0,28
IBk -K 4 -I -A EuclideanDistance	0,89	0,93	0,89	0,3
J48 -C 0,64 -M 2	0,89	0,92	0,89	0,3
RF -I 230 -K 4 -depth 5	0,91	0,94	0,92	0,28
SMO -C 0.96 -N 2 -K "Puk -S 7,8 -O 0,76"	0,91	0,94	0,88	0,3

Desempenho para Demência com discretização e com sobreamostragem



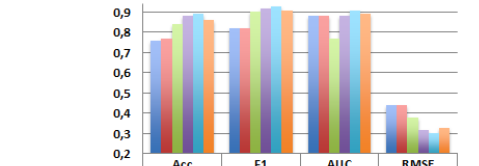
	Acc	F1	AUC	RMSE
BN	0,88	0,92	0,92	0,31
NB	0,88	0,92	0,92	0,32
A1DE	0,89	0,92	0,92	0,28
IBk -K 32 -I -A EuclideanDistance	0,86	0,91	0,9	0,33
J48 -C 0,25 -M 2	0,88	0,92	0,88	0,32
RF -I 209 -K 2 -depth 19	0,87	0,92	0,91	0,3
SMO -C 1,13 -N 0 -K "Puk -S 4,87 -O 0,46"	0,89	0,92	0,9	0,3

Desempenho para Demência sem discretização e sem sobreamostragem



	Acc	F1	AUC	RMSE
LR-NB	0,76	0,82	0,88	0,44
NB	0,76	0,82	0,88	0,45
IBk -K 37 -F -A Euclideandistance	0,84	0,9	0,77	0,39
J48 -U -M 4	0,9	0,93	0,87	0,29
RF -I 100 -k 0 -depth 0	0,9	0,94	0,9	0,3
SMO -C 1.50 -N 1 -K "Puk -S 9,23 -O 0.93"	0,89	0,93	0,86	0,32

Desempenho para Demência sem discretização e com sobreamostragem



	Acc	F1	AUC	RMSE
LR-NB	0,76	0,82	0,88	0,44
NB	0,77	0,82	0,88	0,44
IBk -K 48 -I -A EuclideanDistance	0,84	0,9	0,77	0,38
J48 -U -M 11	0,88	0,92	0,88	0,32
RF -I 100 -k 0 -depth 0	0,89	0,93	0,91	0,3
SMO -C 1,0 -N 0 -K "PolyKernel -E 1"	0,86	0,91	0,89	0,33

Figura 24: Desempenho estimado para o diagnóstico de Demência usando *leave-one-out* para os cenários combinando a aplicação ou não da discretização e da sobreamostragem (testes iniciais).

Tabela 9: Ranqueamento dos classificadores para o diagnóstico de Demência no cenário com discretização e sem sobreamostragem (testes iniciais).

Medida (pontuação)	BN	NB	A1DE	IBk -K 4 -I distância Euclidiana	J48 -C 0,64 -M 2	RF -I 230 -K 4 -depth 5	SMO -C 0,96 -N 2 -K "Puk -S 7,8 -O 0,76"
Acc	0,88 (6)	0,87 (7)	0,9 (3)	0,89 (4,5)	0,89 (4,5)	0,91 (1,5)	0,91 (1,5)
F1	0,92 (5,5)	0,91 (7)	0,93 (3,5)	0,93 (3,5)	0,92 (5,5)	0,94 (1,5)	0,94 (1,5)
AUC	0,92 (3)	0,92 (3)	0,93 (1)	0,89 (5,5)	0,89 (5,5)	0,92 (3)	0,88 (7)
RMSE	0,32 (6,5)	0,32 (6,5)	0,28 (1,5)	0,3 (4)	0,3(4)	0,28 (1,5)	0,3 (4)
Pontuação total	21	23,5	9	17,5	19,5	7,5	14
Posição	6	7	2	4	5	**1**	3

Tabela 10: Ranqueamento dos classificadores para o diagnóstico de Demência no cenário com discretização e com sobreamostragem (testes iniciais).

Medida (pontuação)	BN	NB	A1DE	IBk -K 32 –I distância Euclidiana	J48 -C 0,25 -M 2	RF -I 209 -K 2 -depth 19	SMO -C 1,13 -N 0 -K "Puk -S 4,87 -O 0,46"
Acc	0,88 (4)	0,88 (4)	0,89 (1,5)	0,86 (7)	0,88 (4)	0,87 (6)	0,89 (1,5)
F1	0,92 (3,5)	0,92 (3,5)	0,92 (3,5)	0,91 (7)	0,92 (3,5)	0,92 (3,5)	0,92 (3,5)
AUC	0,92 (2)	0,92 (2)	0,92 (2)	0,9 (5,5)	0,88 (7)	0,91 (4)	0,9 (5,5)
RMSE	0,31 (4)	0,32 (5,5)	0,28 (1)	0,33 (7)	0,32 (5,5)	0,3 (2,5)	0,3 (2,5)
Pontuação total	13,5	15	8	26,5	20	16	13
Posição	3	4	**1**	7	6	5	2

Tabela 11: Ranqueamento dos classificadores para o diagnóstico de Demência no cenário sem discretização e sem sobreamostragem (testes iniciais).

Medida (pontuação)	LR-NB	NB	IBk -K 37 -F distância Euclidiana	J48 –U -M 4	RF -I 100 -k 0 -depth 0	SMO -C 1,50 -N 1 -K "Puk -S 9,23 -O 0,93"
Acc	0,76 (5,5)	0,76 (5,5)	0,84 (4)	0,9 (1,5)	0,9 (1,5)	0,89 (3)
F1	0,82 (5,5)	0,82 (5,5)	0,9 (4)	0,93 (2,5)	0,94 (1)	0,93 (2,5)
AUC	0,88 (2,5)	0,88 (2,5)	0,77 (6)	0,87 (4)	0,9 (1)	0,86 (5)
RMSE	0,44 (5)	0,45 (6)	0,39 (4)	0,29 (1)	0,3 (2)	0,32 (3)
Pontuação total	18,5	19,5	18	9	5,5	13,5
Posição	5	6	4	2	**1**	3

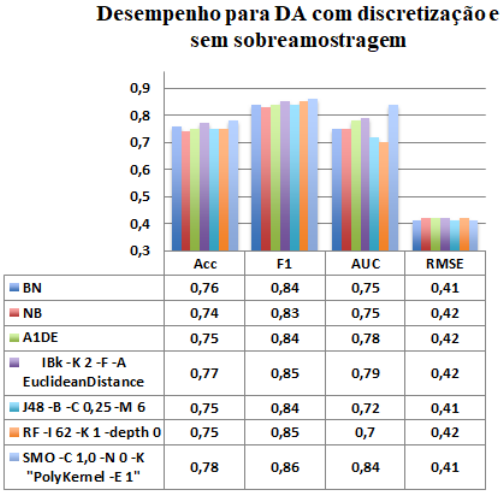
Tabela 12: Ranqueamento dos classificadores para o diagnóstico de Demência no cenário sem discretização e com sobreamostragem (testes iniciais).

Medida (pontuação)	LR-NB	NB	IBk -K 48 -I distância Euclidiana	J48 -U -M 11	RF -I 100 -k 0 -depth 0	SMO -C 1,0 -N 0 -K "PolyKernel -E 1"
Acc	0,76 (6)	0,77 (5)	0,84 (4)	0,88 (2)	0,89 (1)	0,86 (3)
F1	0,82 (5,5)	0,82 (5,5)	0,9 (4)	0,92 (2)	0,93 (1)	0,91 (3)
AUC	0,88 (4)	0,88 (4)	0,77 (6)	0,88 (4)	0,91 (1)	0,89 (2)
RMSE	0,44 (5,5)	0,44 (5,5)	0,38 (4)	0,32 (2)	0,3 (1)	0,33 (3)
Pontuação total	21	20	18	10	4	11
Posição	6	5	4	2	**1**	3

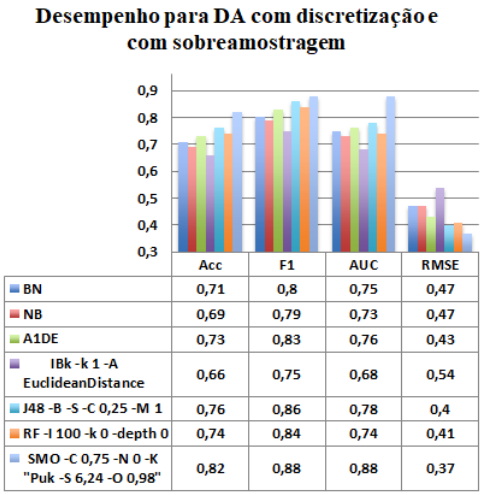
Tabela 13: Escolha do melhor classificador/cenário para diagnóstico de Demência (testes iniciais).

Medida (pontuação)	com discretização sem sobreamostragem RF -I 230 -K 4 -depth 5	com discretização com sobreamostragem A1DE	sem discretização sem sobreamostragem RF -I 100 -k 0 -depth 0	sem discretização com sobreamostragem RF -I 100 -k 0 -depth 0
Acc	0,91 (1)	0,89 (3,5)	0,9 (2)	0,89 (3,5)
F1	0,94 (1,5)	0,92 (4)	0,94 (1,5)	0,93 (3)
AUC	0,92 (1,5)	0,92 (1,5)	0,9 (4)	0,91 (3)
RMSE	0,28 (1,5)	0,28 (1,5)	0,3 (3,5)	0,3 (3,5)
Pontuação total	5,5	10,5	11	13
Posição	**1**	2	3	4

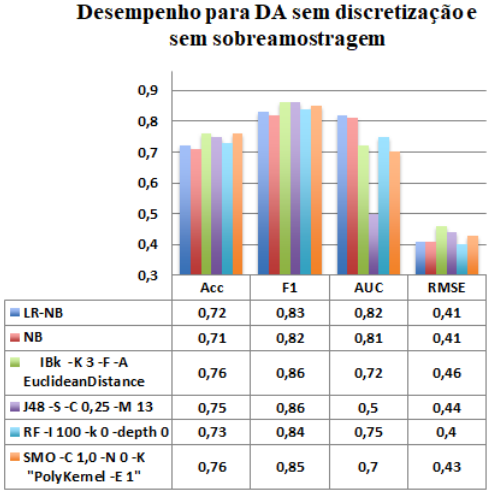
A Figura 25 ilustra os desempenhos estimados para os classificadores avaliados pelo processo de aprendizado para o diagnóstico de DA. Como o conjunto de dados para DA estava desbalanceado (ver Figura 23 (b)), quatro cenários foram avaliados combinando a aplicação ou não da discretização e da sobreamostragem. No cenário sem discretização e com sobreamostragem (ver Tabela 17), os classificadores k-NN e RF empataram, mas o RF foi escolhido por ter prioridade sobre o k-NN, como visto na Seção 4.1.1. Como mostrado nas Tabelas 14-18, o ranqueamento levou a escolha do cenário de pré-processamento com discretização e com sobreamostragem e do classificador SVM com C igual a 0,75 e núcleo universal baseado em função de Pearson VII com sigma igual a 6,24 e ômega igual a 0,98.



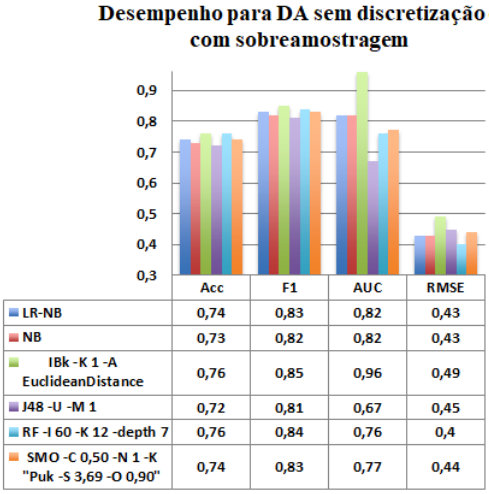
	Acc	F1	AUC	RMSE
BN	0,76	0,84	0,75	0,41
NB	0,74	0,83	0,75	0,42
A1DE	0,75	0,84	0,78	0,42
IBk -K 2 -F -A EuclideanDistance	0,77	0,85	0,79	0,42
J48 -B -C 0,25 -M 6	0,75	0,84	0,72	0,41
RF -I 62 -K 1 -depth 0	0,75	0,85	0,7	0,42
SMO -C 1,0 -N 0 -K "PolyKernel -E 1"	0,78	0,86	0,84	0,41



	Acc	F1	AUC	RMSE
BN	0,71	0,8	0,75	0,47
NB	0,69	0,79	0,73	0,47
A1DE	0,73	0,83	0,76	0,43
IBk -k 1 -A EuclideanDistance	0,66	0,75	0,68	0,54
J48 -B -S -C 0,25 -M 1	0,76	0,86	0,78	0,4
RF -I 100 -k 0 -depth 0	0,74	0,84	0,74	0,41
SMO -C 0,75 -N 0 -K "Puk -S 6,24 -O 0,98"	0,82	0,88	0,88	0,37



	Acc	F1	AUC	RMSE
LR-NB	0,72	0,83	0,82	0,41
NB	0,71	0,82	0,81	0,41
IBk -K 3 -F -A EuclideanDistance	0,76	0,86	0,72	0,46
J48 -S -C 0,25 -M 13	0,75	0,86	0,5	0,44
RF -I 100 -k 0 -depth 0	0,73	0,84	0,75	0,4
SMO -C 1,0 -N 0 -K "PolyKernel -E 1"	0,76	0,85	0,7	0,43



	Acc	F1	AUC	RMSE
LR-NB	0,74	0,83	0,82	0,43
NB	0,73	0,82	0,82	0,43
IBk -K 1 -A EuclideanDistance	0,76	0,85	0,96	0,49
J48 -U -M 1	0,72	0,81	0,67	0,45
RF -I 60 -K 12 -depth 7	0,76	0,84	0,76	0,4
SMO -C 0,50 -N 1 -K "Puk -S 3,69 -O 0,90"	0,74	0,83	0,77	0,44

Figura 25: Desempenho estimado para o diagnóstico de DA usando *leave-one-out* para os cenários combinando a aplicação ou não da discretização e da sobreamostragem (testes iniciais).

Tabela 14: Ranqueamento dos classificadores para o diagnóstico de DA no cenário com discretização e sem sobreamostragem (testes iniciais).

Medida (pontuação)	BN	NB	A1DE	IBk -K 2 -F distância Euclidiana	J48 –B -C 0,25 -M 6	RF -I 62 -K 1 -depth 0	SMO -C 1,0 -N 0 -K "PolyKernel -E 1"
Acc	0,76 (3)	0,74 (7)	0,75 (5)	0,77 (2)	0,75 (5)	0,75 (5)	0,78 (1)
F1	0,84 (5)	0,83 (7)	0,84 (5)	0,85 (2,5)	0,84 (5)	0,85 (2,5)	0,86 (1)
AUC	0,75 (4,5)	0,75 (4,5)	0,78 (3)	0,79 (2)	0,72 (6)	0,7 (7)	0,84 (1)
RMSE	0,41 (2)	0,42 (5,5)	0,42 (5,5)	0,42 (5,5)	0,41 (2)	0,42 (5,5)	0,41 (2)
Pontuação total	14,5	24	18,5	12	18	20	5
Posição	3	6	5	2	4	6	**1**

Tabela 15: Ranqueamento dos classificadores para o diagnóstico de DA no cenário com discretização e com sobreamostragem (testes iniciais).

Medida (pontuação)	BN	NB	A1DE	IBk -K 1 distância Euclidiana	J48 -B -S -C 0,25 -M 1	RF -I 100 -k 0 -depth 0	SMO -C 0,75 -N 0 -K "Puk -S 6,24 -O 0,98"
Acc	0,71 (5)	0,69 (6)	0,73 (4)	0,66 (7)	0,76 (2)	0,74 (3)	0,82 (1)
F1	0,8 (5)	0,79 (6)	0,83 (4)	0,75 (7)	0,86 (2)	0,84 (3)	0,88 (1)
AUC	0,75 (4)	0,73 (6)	0,76 (3)	0,68 (7)	0,78 (2)	0,74 (5)	0,88 (1)
RMSE	0,47 (5,5)	0,47 (5,5)	0,43 (4)	0,54 (7)	0,4 (2)	0,41 (3)	0,37 (1)
Pontuação total	19,5	23,5	15	28	8	14	4
Posição	5	6	4	7	2	3	**1**

Tabela 16: Ranqueamento dos classificadores para o diagnóstico de DA no cenário sem discretização e sem sobreamostragem (testes iniciais).

Medida (pontuação)	LR-NB	NB	IBk -K 3 -F distância Euclidiana	J48 -S -C 0,25 -M 13	RF -I 100 -k 0 -depth 0	SMO -C 1,0 -N 0 -K "PolyKernel -E 1"
Acc	0,72 (5)	0,71 (6)	0,76 (1,5)	0,75 (3)	0,73 (4)	0,76 (1,5)
F1	0,83 (5)	0,82 (6)	0,86 (1,5)	0,86 (1,5)	0,84 (4)	0,85 (3)
AUC	0,82 (1)	0,81 (2)	0,72 (4)	0,5 (6)	0,75 (3)	0,7 (5)
RMSE	0,41 (2,5)	0,41 (2,5)	0,46 (6)	0,44 (5)	0,4 (1)	0,43 (4)
Pontuação total	13,5	16,5	13	15,5	12	13,5
Posição	3,5	6	2	5	**1**	3,5

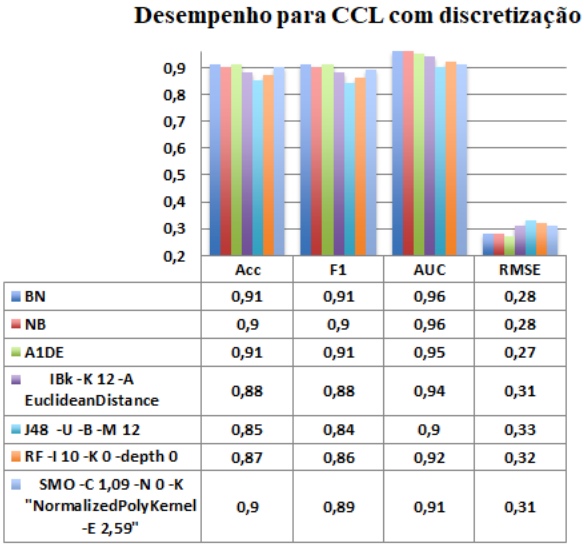
Tabela 17: Ranqueamento dos classificadores para o diagnóstico de DA no cenário sem discretização e com sobreamostragem (testes iniciais).

Medida (pontuação)	LR-NB	NB	IBk -K 1 distância Euclidiana	J48 -U -M 1	RF -I 60 -K 12 -depth 7	SMO -C 0,50 -N 1 -K "Puk -S 3,69 -O 0,90"
Acc	0,74 (3,5)	0,73 (5)	0,76 (1,5)	0,72 (6)	0,76 (1,5)	0,74 (3,5)
F1	0,83 (3,5)	0,82 (5)	0,85 (1)	0,81 (6)	0,84 (2)	0,83 (3,5)
AUC	0,82 (2,5)	0,82 (2,5)	0,96 (1)	0,67 (6)	0,76 (5)	0,77 (4)
RMSE	0,43 (2,5)	0,43 (2,5)	0,49 (6)	0,45 (5)	0,4 (1)	0,44 (4)
Pontuação total	12	15	9,5	23	9,5	15
Posição	3	4,5	**1,5**	6	**1,5**	4,5

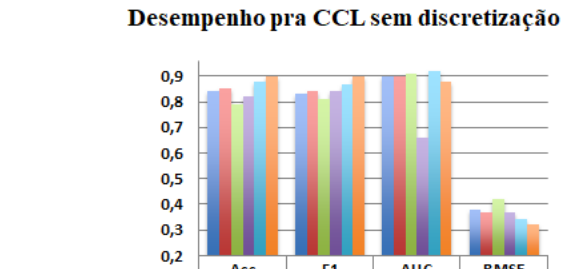
Tabela 18: Escolha do melhor classificador/cenário para diagnóstico de DA (testes iniciais).

Medida (pontuação)	com discretização sem sobreamostragem SMO -C 1,0 -N 0 -K "PolyKernel -E 1"	com discretização com sobreamostragem SMO -C 0,75 -N 0 -K "Puk -S 6,24 -O 0,98"	sem discretização sem sobreamostragem RF -I 100 -k 0 -depth 0	sem discretização com sobreamostragem RF -I 60 -K 12 -depth 7
Acc	0,78 (2)	0,82 (1)	0,73 (4)	0,76 (3)
F1	0,86 (2)	0,88 (1)	0,84 (3,5)	0,84 (3,5)
AUC	0,84 (2)	0,88 (1)	0,75 (4)	0,76 (3)
RMSE	0,41 (4)	0,37 (1)	0,4 (2,5)	0,4 (2,5)
Pontuação total	10	4	14	12
Posição	2	**1**	4	3

A Figura 26 ilustra os desempenhos estimados para os classificadores avaliados pelo processo de aprendizado para o diagnóstico de CCL. Como o conjunto de dados para CCL estava balanceado (ver Figura 23 (c)), somente dois cenários de pré-processamento foram avaliados: com e sem discretização. No cenário com discretização (ver Tabela 19), os classificadores BN e A1DE empataram, mas o BN foi escolhido por ter prioridade sobre o A1DE, como visto na Seção 4.1.1. Como mostrado nas Tabelas 19-21, o processo de ranqueamento levou a escolha do classificador BN no cenário com discretização.



	Acc	F1	AUC	RMSE
BN	0,91	0,91	0,96	0,28
NB	0,9	0,9	0,96	0,28
A1DE	0,91	0,91	0,95	0,27
IBk -K 12 -A EuclideanDistance	0,88	0,88	0,94	0,31
J48 -U -B -M 12	0,85	0,84	0,9	0,33
RF -I 10 -K 0 -depth 0	0,87	0,86	0,92	0,32
SMO -C 1,09 -N 0 -K "NormalizedPolyKernel -E 2,59"	0,9	0,89	0,91	0,31



	Acc	F1	AUC	RMSE
LR-NB	0,84	0,83	0,9	0,38
NB	0,85	0,84	0,9	0,37
IBk -E -K 18 EuclideanDistance	0,79	0,81	0,91	0,42
J48 -C 0,25 -M 19	0,82	0,84	0,66	0,37
RF -I 22 -K 0 -depth 0	0,88	0,87	0,92	0,34
SMO -C 0,90 -N 0 -K "NormalizedPolyKernel -E 3 -L"	0,9	0,9	0,88	0,32

Figura 26: Desempenho estimado para o diagnóstico de CCL usando *leave-one-out* para os cenários com e sem discretização (testes iniciais).

Tabela 19: Ranqueamento dos classificadores para o diagnóstico de CCL no cenário com discretização (testes iniciais).

Medida (pontuação)	BN	NB	A1DE	IBk -K 12 distância Euclidiana	J48 -U -B -M 12	RF -I 10 -K 0 -depth 0	SMO -C 1,09 -N 0 -K "NormalizedPoly Kernel -E 2,59"
Acc	0,91 (1,5)	0,9 (3,5)	0,91 (1,5)	0,88 (5)	0,85 (7)	0,87 (6)	0,9 (3,5)
F1	0,91 (1,5)	0,9 (3)	0,91 (1,5)	0,88 (5)	0,84 (7)	0,86 (6)	0,89 (4)
AUC	0,96 (1,5)	0,96 (1,5)	0,95 (3)	0,94 (4)	0,9 (7)	0,92 (5)	0,91 (6)
RMSE	0,28 (2,5)	0,28 (2,5)	0,27 (1)	0,31 (4,5)	0,33 (7)	0,32 (6)	0,31 (4,5)
Pontuação total	7	23,5 (10,5)	7	17,5 (18,5)	28	23	18
Posição	**1,5**	3	**1,5**	5	7	6	4

Tabela 20: Ranqueamento dos classificadores para o diagnóstico de CCL no cenário sem discretização (testes iniciais).

Medida (pontuação)	LR-NB	NB	IBk -E-K 18 distância Euclidiana	J48 -C 0,25 -M 19	RF -I 22 -K 0 -depth 0	SMO -C 0.90 -N 0 -K "NormalizedPoly Kernel -E 3 -L"
Acc	0,84 (4)	0,85 (3)	0,79 (6)	0,82 (5)	0,88 (2)	0,9 (1)
F1	0,83 (5)	0,84 (3,5)	0,81 (6)	0,84 (3,5)	0,87 (2)	0,9 (1)
AUC	0,9 (3,5)	0,9 (3,5)	0,91 (2)	0,66 (6)	0,92 (1)	0,88 (5)
RMSE	0,38 (5)	0,37 (3,5)	0,42 (6)	0,37 (3,5)	0,34 (2)	0,32 (1)
Pontuação total	17,5	13,5	20	18	7	8
Posição	4	3	6	5	**1**	2

Tabela 21: Escolha do melhor classificador/cenário para diagnóstico de CCL (testes iniciais).

Medida (pontuação)	com discretização BN	sem discretização RF -I 22 -K 0 -depth 0
Acc	0,91 (1)	0,88 (2)
F1	0,91 (1)	0,87 (2)
AUC	0,96 (1)	0,92 (2)
RMSE	0,28 (1)	0,34 (2)
Pontuação total	4	8
Posição	**1**	2

Note que, para Demência, a discretização supervisionada dos atributos contínuos contribuiu para a melhoria do desempenho geral dos classificadores quando a

sobreamostragem não foi aplicada (ver Figura 24). Embora a sobreamostragem sobre o conjunto de dados desbalanceado de DA apenas tenha contribuído para melhoria do desempenho geral dos classificadores levando-se em consideração a AUC, quando a discretização não foi aplicada (ver Figura 25), o cenário escolhido foi o com discretização e com sobreamostragem para o classificador SVM que alcançou o melhor desempenho quando comparado aos classificadores vencedores dos demais cenários (ver Tabela 18). Para CCL, a discretização supervisionada dos atributos contínuos contribuiu para a melhoria do desempenho geral dos classificadores (ver Figura 26).

6.2.2.2 ATRIBUTOS DOS CLASSIFICADORES BINÁRIOS OBTIDOS

As Tabelas 22-24 mostram os atributos selecionados para os classificadores binários obtidos respectivamente para o diagnóstico de Demência, DA e CCL após o processo de aprendizado automático utilizando os dados do CDA para treinamento. Tais tabelas também mostram os intervalos de discretização obtidos para os atributos contínuos que foram discretizados.

Tabela 22: Lista de atributos selecionados para diagnóstico de Demência (testes iniciais).

Atributo	**Valores possíveis**[20]
Idade	1:[0-72]; 2:[73-inf]
Gênero	1:feminino; 2:masculino
Escolaridade	1:[0-13]; 2:[14-inf)
MMSE	1:[0-17]; 2:[18-26]; 3:[27-30]
CDR	1:0; 2:0,5; 3:1; 4:2; 5:3
VFT	1:[0-4]; 2:[5-11]; 3:[12-inf]
TMT – Parte A	1:[0-16]; 2:[17-59]; 3:[60-inf]
Stroop	1:[0-15]; 2:[16-inf)
Lawton	1:[9-25]; 2:[26-27]
CDT	1:0; 2:1; 3:2; 4:3; 5:4; 6:5
Pfeffer	1:0; 2:[1-2]; 3:[3-30]
IQCODE	1:[1-3,35]; 2:[3,36–5]
Berg	1:[0-51]; 2:[52-56]
Depressão	1:ausência; 2:presença

Tabela 23: Lista de atributos selecionados para diagnóstico de DA (testes iniciais).

Atributo	**Valores possíveis**
Idade	1:[0-70]; 2:[71-76]; 3:[77-81]; 4:[82:inf]
Gênero	1:feminino; 2:masculino
Escolaridade	1:[0-3]; 2:[4-5]; 3:[6-7]; 4:[8-inf]

[20]Representa os valores discretos que o atributo pode assumir. A sintaxe para atributos que foram discretizados é n*:[J–K]*, onde n=valor, *J*=limite inferior do intervalo; *K*=limite superior do intervalo; *inf*=infinito. A discretização supervisionada estimou tais limites.

MMSE	1:[0-10]; 2:[11-16];3:[17-21]; 4:[22-30]
CDR	1:0; 2:0,5; 3:1; 4:2; 5:3
VFT	1:[0-3]; 2:[4-7]; 3:[8-10]; 4:[11-inf]
TMT- Parte A	1:[0-22]; 2:[23-89]; 3:[90-140]; 4:[141-inf]
Stroop	1:[0-21]; 2:[22-31]; 3:[32-43]; 4:[44-inf]
Lawton	1:[9-14]; 2:[15-19]; 3:[20-22]; 4:[23-27]
CDT	1:0; 2:1; 3:2; 4:3; 5:4; 6:5
Pfeffer	1:[0-11]; 2:[12-20]; 3:[21-27]; 4:[28-30]
IQCODE	1:[1-3,46]; 2:[3,47-3,97]; 3:[3,98-4,41]; 4:[4,42-5]
Berg	1:[0-40]; 2:[41-46]; 3:[47-51]; 4:[52-56]
Depressão	1:ausência; 2:presença

Tabela 24: Lista de atributos selecionados para diagnóstico de CCL (testes iniciais).

Atributo	**Valores possíveis**
Gênero	1:feminino; 2:masculino
Escolaridade	1:[0-15]; 2:[16-inf]
MMSE	1:[0-28]; 2:[29-30]
CDR	1:0; 2:0,5; 3:1; 4:2; 5:3
VFT	1:[0-15]; 2:[16-inf]
TMT – Parte A	1:[0-36]; 2:[37-inf]
Stroop	1:[0-17]; 2:[18-inf]
Lawton	1:[9-24]; 2:[25-27]
CDT	1:0; 2:1; 3:2; 4:3; 5:4; 6:5
Pfeffer	1:[0-1]; 2:[2-inf]
IQCODE	1:[1-3,32]; 2:[3,33-5]
Berg	1:[0-54]; 2:[55]; 3:[56]
Depressão	1:ausência; 2:presença

6.2.3 TESTES COM DADOS NÃO UTILIZADOS NO TREINAMENTO

As Tabelas 25-27 mostram as matrizes confusão obtidas usando os dados do CRASI (ver Figura 23 (d-f)) para testar os classificadores automaticamente obtidos para Demência, DA e CCL (respectivamente RF -I 230 -K 4 -depth 5, SMO -C 0,75 -N 0 -K "Puk -S 6,24 -O 0,98" e BN). Para a realização dos testes, exames presentes nas bases de testes que não estavam presentes nos classificadores obtidos não foram considerados. Exames presentes apenas nos classificadores obtidos, mas não nas bases de testes, foram considerados dados não informados.

Para Demência, o classificador acertou a classificação para quase todas as tuplas de teste, apresentando apenas 1 falso positivo, com Acc de 98,6%, F1 de 0,96, AUC de 0,999 e RMSE de 0,26. Portanto, o desempenho para Demência foi melhor que o estimado pelo *leave-one-out*: Acc de 91%, F1 de 0,94, AUC de 0,92 e RMSE de 0,28 (ver Tabela 13).

Para DA, apenas 4 tuplas foram testadas (ver Figura 23 (e)). O classificador acertou a classificação para todas as tuplas de teste classificando-as corretamente como positivas.

Para CCL, os resultados para a base com dados de teste em separado foram: Acc de 68%, F1 de 0,73, AUC de 0,96 e RMSE de 0,46. Portanto, o desempenho para CCL foi pior do que o estimado pelo *leave-one-out*: Acc de 91%, F1 de 0,91, AUC de 0,96 e RMSE de 0,28 (ver Tabela 21). Ressalte-se o grande número de falsos negativos: 19.

Tabela 25: Matriz confusão obtida para Demência (testes iniciais).

Predito	***Ground Truth***	
	Negativo	**Positivo**
Negativo	59	0
Positivo	1	12

Tabela 26: Matriz confusão obtida para DA (testes iniciais).

Predito	***Ground Truth***	
	Negativo	**Positivo**
Negativo	0	0
Positivo	0	4

Tabela 27: Matriz confusão obtida para CCL (testes iniciais).

Predito	***Ground Truth***	
	Negativo	**Positivo**
Negativo	15	19
Positivo	0	26

6.3 TESTES FINAIS UTILIZANDO DADOS DO CDA E DO CRASI PARA CONSTRUIR O MODELO DE DECISÃO E DADOS DO CRASI PARA TESTAR A GENERALIZAÇÃO DO MODELO OBTIDO

6.3.1 CONJUNTOS DE DADOS UTILIZADOS NOS TESTES FINAIS

Nos testes finais, os dados do CDA e os dados do CRASI utilizados nos testes iniciais foram usados para testar o processo automático que constrói o modelo de decisão escolhendo o melhor classificador e cenário de pré-processamento para cada doença/transtorno. Por outro lado, dados adicionais do CRASI, coletados posteriormente, foram utilizados para testar os classificadores obtidos com dados novos não utilizados no treinamento. A Figura 27 (a-c) mostra o número de pacientes com diagnóstico confirmado (positivo ou negativo) do CDA e do CRASI cujos dados foram usados para testar o processo automático de aprendizado e a Figura 27 (d-f) mostra o número de pacientes com diagnóstico confirmado do CRASI cujos dados foram usados para testar o modelo de decisão obtido formando conjuntos de teste em separado. De acordo com o processo diagnóstico ilustrado pela Figura 15, dados de pacientes diagnosticados como dementes podem compor os conjuntos de dados de DA enquanto dados

de pacientes diagnosticados como não dementes podem compor os conjuntos de dados de CCL. Note-se que, dos 26 pacientes diagnosticados como positivos para Demência no CRASI (ver Figura 27 (d)), apenas 11 pacientes passaram por todo o processo diagnóstico sendo diagnosticados quanto à Demência ser do tipo DA ou não (ver Figura 27 (e)). Os outros 15 pacientes foram diagnosticados como positivos para Demência de todos os tipos, sem diagnóstico se a Demência é devida à DA ou não.

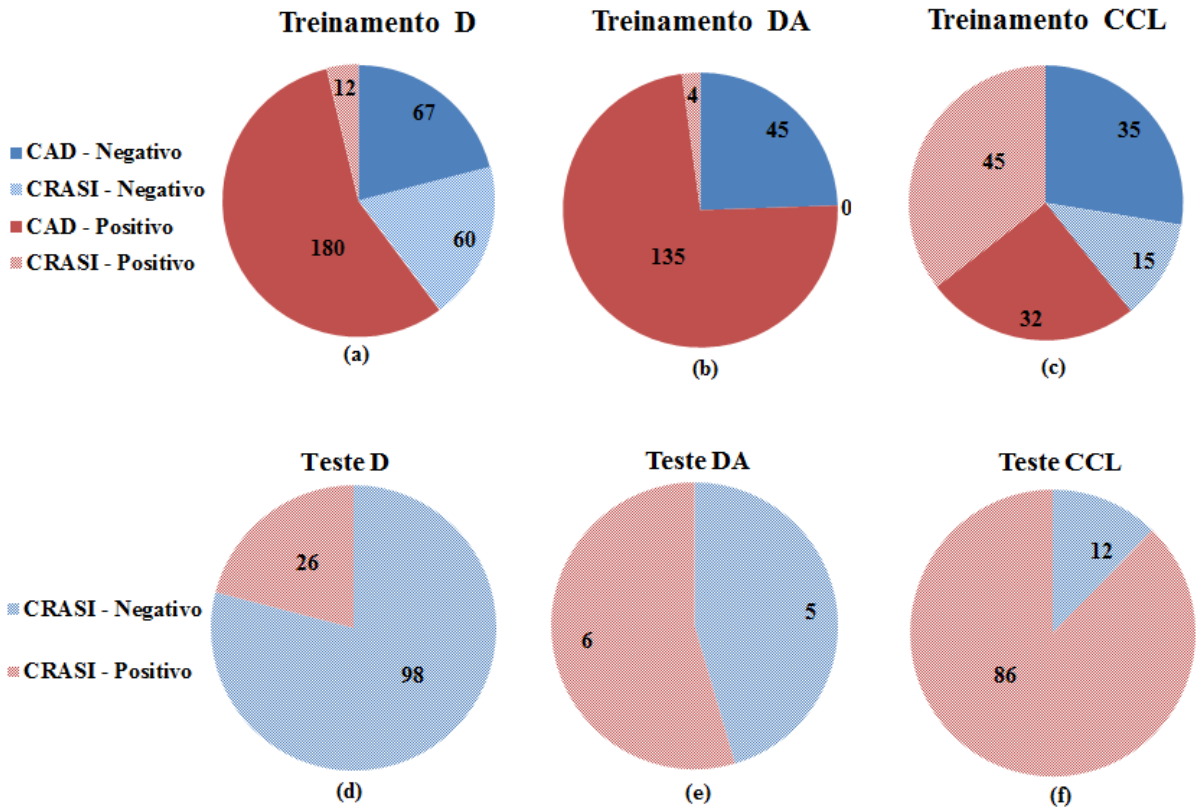


Figura 27: Número de pacientes diagnosticados nos conjuntos de dados para Demência, DA e CCL usados nos testes finais para treinamento (a-c) e teste (d-f).

6.3.2 RESULTADOS ALCANÇADOS PELO PROCESSO AUTOMÁTICO DE APRENDIZADO DO MODELO

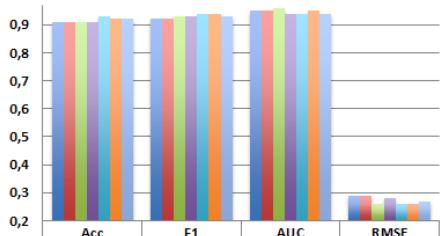
O método *learn* foi executado usando dados rotulados do CDA e do CRASI (ver Figura 27 (a-c)) e levou 145 minutos para ser executado em *background* por um processador Intel de 1.6 GHz Intel com memória de 4GB. A maior parte deste tempo (cerca de 90%) foi gasto pela otimização de hiperparâmetros realizada pela API AutoWEKA. Todos os resultados de desempenho estimado e ranqueamento dos classificadores foram registrados pelo processo, sendo apresentados na próxima seção. Ressalte-se que o tempo de execução total dos testes finais foi menor do que o dos testes iniciais, porque, os testes iniciais

avaliaram um maior número de cenários de pré-processamento, decorrência da base de Demência estar desbalanceada.

6.3.2.1 DESEMPENHO ESTIMADO PARA OS CLASSIFICADORES E CENÁRIOS AVALIADOS E SELEÇÃO DO MELHOR CLASSIFICADOR E CENÁRIO PARA DEMÊNCIA, DA E CCL

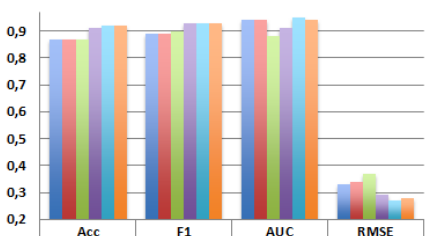
A Figura 28 ilustra os desempenhos estimados para os classificadores avaliados pelo processo de aprendizado para o diagnóstico de Demência. RMSE é a única medida em que valores mais próximos a zero indicam melhor desempenho. Para as demais medidas, valores mais próximos a 1 indicam melhor desempenho. Para cada cenário de pré-processamento, hiperparâmetros específicos foram escolhidos pela AutoWEKA (ver Tabela 7). Como o conjunto de dados de Demência estava balanceado (ver Figura 27 (a), união dos dados de CDA e CRASI), somente dois cenários de pré-processamento foram avaliados: com e sem discretização. As Tabelas 28-29 mostram os resultados do processo de ranqueamento que foi realizado para os dois cenários. A Tabela 30 mostra o ranqueamento realizado para o melhor classificador de cada cenário e a escolha do melhor cenário de pré-processamento e classificador. Para Demência, o vencedor foi o cenário com discretização e o classificador RF com número de árvores igual a 7, máxima profundidade das árvores igual a 4 e número de atributos aleatoriamente escolhidos igual a $\text{int}\,(\log_2(\text{número total de atributos} + 1))$, que resulta em 4, pois o número obtido de atributos para o classificador de Demência foi 13 como mostrado na Seção 6.3.2.2.

Desempenho para D com discretização



	Acc	F1	AUC	RMSE
BN	0,91	0,92	0,95	0,29
NB	0,91	0,92	0,95	0,29
A1DE	0,91	0,93	0,96	0,26
IBk -k 10 -A EuclideanDistance	0,91	0,93	0,94	0,28
J48 -C 0,25 -M 2	0,93	0,94	0,94	0,26
RF -I 7 -K 0 -depth 4	0,92	0,94	0,95	0,26
SMO -C 1 -N 0 -K "NormalizedPolyKernel -E 1"	0,92	0,93	0,94	0,27

Desempenho para D sem discretização



	Acc	F1	AUC	RMSE
LR-NB	0,87	0,89	0,94	0,33
NB	0,87	0,89	0,94	0,34
IBk -k 10 -A EuclideanDistance	0,87	0,9	0,88	0,37
J48 -S -C 0,25 -M 13	0,91	0,93	0,91	0,29
RF -I 100 -K 0 -depth 0	0,92	0,93	0,95	0,27
SMO -C 0,69 -N 1 -K "RBFKernel -G 0,0031"	0,92	0,93	0,94	0,28

Figura 28: Desempenho estimado para o diagnóstico de Demência usando *leave-one-out* para os cenários com e sem discretização (testes finais).

Tabela 28: Ranqueamento dos classificadores para o diagnóstico de Demência no cenário com discretização (testes finais).

Medida (pontuação)	BN	NB	A1DE	IBk -k 10 distância Euclidiana	J48 -C 0,25 -M 2	RF -I 7 -K 0 -depth 4	SMO -C 1 -N 0 -K "Normalized PolyKernel -E 1"
Acc	0,91 (5,5)	0,91 (5,5)	0,91 (5,5)	0,91 (5,5)	0,93 (1)	0,92 (2,5)	0,92 (2,5)
F1	0,92 (6,5)	0,92 (6,5)	0,93 (4)	0,93 (4)	0,94 (1,5)	0,94 (1,5)	0,93 (4)
AUC	0,95 (3)	0,95 (3)	0,96 (1)	0,94 (6)	0,94 (6)	0,95 (3)	0,94 (6)
RMSE	0,29 (6,5)	0,29 (6,5)	0,26 (2)	0,28 (5)	0,26 (2)	0,26 (2)	0,27 (4)
Pontuação total	21,5	21,5	12,5	20,5	10,5	9	16,5
Posição	6,5	6,5	3	5	2	**1**	4

Tabela 29: Ranqueamento dos classificadores para o diagnóstico de Demência no cenário sem discretização (testes finais).

Medida (pontuação)	LR-NB	NB	IBk -k 10 distância Euclidiana	J48 -S -C 0,25 -M 13	RF -I 100 -K 0 -depth 0	SMO -C 0,69 -N 1 -K "RBFKernel -G 0,0031"
Acc	0,87 (5)	0,87 (5)	0,87 (5)	0,91 (3)	0,92 (1,5)	0,92 (1,5)
F1	0,89 (5,5)	0,89 (5,5)	0,90 (4)	0,93 (2)	0,93 (2)	0,93 (2)
AUC	0,94 (3)	0,94 (3)	0,88 (6)	0,91 (5)	0,95 (1)	0,94 (3)
RMSE	0,33 (4)	0,34 (5)	0,37 (6)	0,29 (3)	0,27 (1)	0,28 (2)
Pontuação total	17,5	18,5	21	13	5,5	8,5
Posição	4	5	6	3	**1**	2

Tabela 30: Escolha do melhor classificador/cenário para diagnóstico de Demência (testes finais).

Medida (pontuação)	com discretização RF -I 7 -K 0 -depth 4	sem discretização RF -I 100 -K 0 -depth 0
Acc	0,92 (1,5)	0,92 (1,5)
F1	0,94 (1)	0,93 (2)
AUC	0,95 (1,5)	0,95 (1,5)
RMSE	0,26 (1)	0,27 (2)
Pontuação total	5	7
Posição	**1**	2

A Figura 29 ilustra os desempenhos estimados para os classificadores avaliados pelo processo de aprendizado para o diagnóstico de DA. Como o conjunto de dados para DA estava desbalanceado (ver Figura 27 (b), união de CDA e CRASI), quatro cenários foram avaliados combinando a aplicação ou não da discretização e da sobreamostragem. Como

mostrado nas Tabelas 31-35, o ranqueamento levou a escolha do cenário de pré-processamento com discretização e sem sobreamostragem e do classificador SVM com C igual a 0,77, núcleo polinomial com expoente 1,26 e uso de termos de ordem mais baixa.

Desempenho para DA com discretização e sem sobreamostragem	Acc	F1	AUC	RMSE
BN	0,73	0,84	0,77	0,41
NB	0,73	0,84	0,78	0,41
A1DE	0,72	0,83	0,77	0,41
IBk -k 6 -I -A EuclideanDistance	0,72	0,82	0,77	0,4
J48 -C 0,25 -M 2	0,76	0,86	0,5	0,43
RF -I 73 -K 1 -depth 2	0,76	0,86	0,72	0,42
SMO -C 0,77 -N 2 -K "PolyKernel -E 1,26 -L"	0,85	0,9	0,89	0,37

Desempenho para DA com discretização e com sobreamostragem	Acc	F1	AUC	RMSE
BN	0,68	0,78	0,74	0,47
NB	0,7	0,79	0,72	0,46
A1DE	0,74	0,83	0,77	0,43
IBk -k 5 -F -A EuclideanDistance	0,67	0,75	0,7	0,48
J48 -B -C 0,33 -M 10	0,74	0,84	0,66	0,43
RF -I 62 -K 1 -depth 0	0,74	0,85	0,73	0,41
SMO -C 0,69 -N 2 -K "Puk -S 0,99 -O 0,59"	0,79	0,86	0,84	0,43

Desempenho para DA sem discretização e sem sobreamostragem	Acc	F1	AUC	RMSE
LR-NB	0,74	0,84	0,82	0,4
NB	0,73	0,84	0,82	0,4
IBk -K 39 -F -A EuclideanDistance	0,76	0,86	0,62	0,42
J48 -B -S -C 0,45 -M 29	0,76	0,86	0,5	0,43
RF I 18 -K 0 -depth 20	0,76	0,86	0,69	0,41
SMO -C 1,21 -N 2 -K "RBFKernel -G 0,016"	0,77	0,86	0,66	0,45

Desempenho para DA sem discretização e com sobreamostragem	Acc	F1	AUC	RMSE
LR-NB	0,79	0,86	0,86	0,39
NB	0,78	0,86	0,86	0,39
IBk -K 39 -F -A EuclideanDistance	0,78	0,86	0,7	0,41
J48 -C 0,25 -M 2	0,77	0,85	0,61	0,43
RF -I 209 -K 2 -depth 19	0,78	0,86	0,75	0,4
SMO -C 0,50 -N 1 -K "Puk -S 3,69 -O 0,90"	0,78	0,86	0,76	0,42

Figura 29: Desempenho estimado para o diagnóstico de DA usando *leave-one-out* para os cenários combinando a aplicação ou não da discretização e da sobreamostragem (testes finais).

Tabela 31: Ranqueamento dos classificadores para o diagnóstico de DA no cenário com discretização e sem sobreamostragem (testes finais).

Medida (pontuação)	BN	NB	A1DE	IBk -k 6 -I distância Euclidiana	J48 -C 0,25 -M 2	RF -I 73 -K 1 -depth 2	SMO -C 0,77 -N 2 -K "PolyKernel -E 1,26 -L"
Acc	0,73 (4,5)	0,73 (4,5)	0,72 (6,5)	0,72 (6,5)	0,76 (2,5)	0,76 (2,5)	0,85 (1)
F1	0,84 (4,5)	0,84 (4,5)	0,83 (6)	0,82 (7)	0,86 (2,5)	0,86 (2,5)	0,9 (1)
AUC	0,77 (4)	0,78 (2)	0,77 (4)	0,77 (4)	0,5 (7)	0,72 (6)	0,89 (1)
RMSE	0,41 (4)	0,41 (4)	0,41 (4)	0,4 (2)	0,43 (7)	0,42 (6)	0,37 (1)
Pontuação total	17	15	20,5	19,5	19	17	4
Posição	3,5	2	7	6	5	3,5	**1**

Tabela 32: Ranqueamento dos classificadores para o diagnóstico de DA no cenário com discretização e com sobreamostragem (testes finais).

Medida (pontuação)	BN	NB	A1DE	IBk -k 5 -F distância Euclidiana	J48 –B -C 0,33 -M 10	RF -I 62 -K 1 -depth 0	SMO -C 0,69 -N 2 -K "Puk -S 0,99 -O 0,59"
Acc	0,68 (6)	0,7 (5)	0,74 (3)	0,67 (7)	0,74 (3)	0,74 (3)	0,79 (1)
F1	0,78 (6)	0,79 (5)	0,83 (4)	0,75 (7)	0,84 (3)	0,85 (2)	0,86 (1)
AUC	0,74 (3)	0,72 (5)	0,77 (2)	0,7 (6)	0,66 (7)	0,73 (4)	0,84 (1)
RMSE	0,47 (6)	0,46 (5)	0,43 (3)	0,48 (7)	0,43 (3)	0,41 (1)	0,43 (3)
Pontuação total	21	20	12	27	16	10	6
Posição	6	5	3	7	4	2	**1**

Tabela 33: Ranqueamento dos classificadores para o diagnóstico de DA no cenário sem discretização e sem sobreamostragem (testes finais).

Medida (pontuação)	LR-NB	NB	IBk -K 39 -F distância Euclidiana	J48 -B –S -C 0,45 -M 29	RF -I 18 -K 0 -depth 20	SMO -C 1,21 -N 2 -K "RBFKernel -G 0,016"
Acc	0,74 (5)	0,73 (6)	0,76 (3)	0,76 (3)	0,76 (3)	0,77 (1)
F1	0,84 (5,5)	0,84 (5,5)	0,86 (2,5)	0,86 (2,5)	0,86 (2,5)	0,86 (2,5)
AUC	0,82 (1,5)	0,82 (1,5)	0,62 (5)	0,5 (6)	0,69 (3)	0,66 (4)
RMSE	0,4 (1,5)	0,4 (1,5)	0,42 (4)	0,43 (5)	0,41 (3)	0,45 (6)
Pontuação total	13,5	14,5	14,5	16,5	11,5	13,5
Posição	2,5	4,5	4,5	6	**1**	2,5

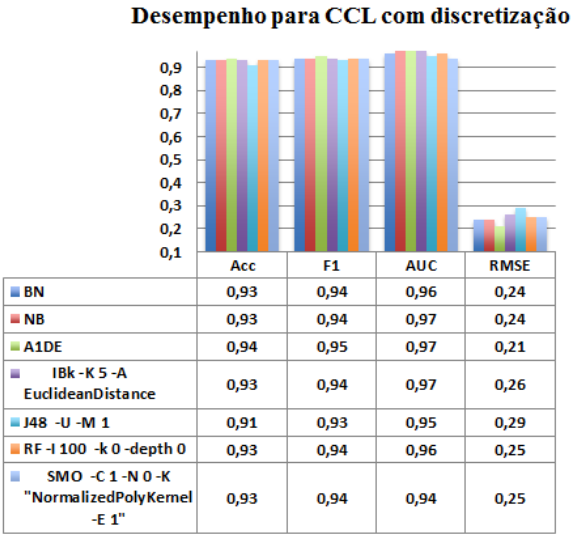
Tabela 34: Ranqueamento dos classificadores para o diagnóstico de DA no cenário sem discretização e com sobreamostragem (testes finais).

Medida (pontuação)	LR-NB	NB	IBk -K 39 -F distância Euclidiana	J48 -C 0,25 -M 2	RF -I 209 -K 2 -depth 19	SMO -C 0,50 -N 1 -K "Puk -S 3,69 -O 0,90"
Acc	0,79 (1)	0,78 (3,5)	0,78 (3,5)	0,77 (6)	0,78 (3,5)	0,78 (3,5)
F1	0,86 (3)	0,86 (3)	0,86 (3)	0,85 (6)	0,86 (3)	0,86 (3)
AUC	0,86 (1,5)	0,86 (1,5)	0,7 (5)	0,61 (6)	0,75 (4)	0,76 (3)
RMSE	0,39 (1,5)	0,39 (1,5)	0,41 (4)	0,43 (6)	0,4 (3)	0,42 (5)
Pontuação total	7	9,5	15,5	24	13,5	14,5
Posição	**1**	2	5	6	3	4

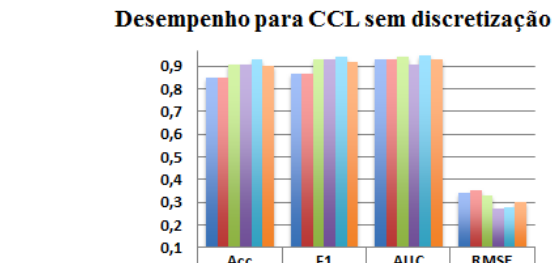
Tabela 35: Escolha do melhor classificador e cenário para diagnóstico de DA (testes finais).

Medida (pontuação)	com discretização sem sobreamostragem SMO -C 0,77 -N 2 -K "PolyKernel -E 1,26 -L"	com discretização com sobreamostragem SMO -C 0,69 -N 2 -K "Puk -S 0,99 -O 0,59"	sem discretização sem sobreamostragem RF I 18 -K 0 -depth 20	sem discretização com sobreamostragem LR-NB
Acc	0,85 (1)	0,79 (2,5)	0,76 (4)	0,79 (2,5)
F1	0,9 (1)	0,86 (3)	0,86 (3)	0,86 (3)
AUC	0,89 (1)	0,84 (3)	0,69 (4)	0,86 (2)
RMSE	0,37 (1)	0,43 (4)	0,41 (3)	0,39 (2)
Pontuação total	4	12,5	14	9,5
Posição	**1**	3	4	2

A Figura 30 ilustra os desempenhos estimados para os classificadores avaliados pelo processo de aprendizado para o diagnóstico de CCL. Como o conjunto de dados para CCL estava balanceado (ver Figura 27 (c), união de CDA e CRASI), somente dois cenários de pré-processamento foram avaliados: com e sem discretização. Como mostrado nas Tabelas 36-38, o processo de ranqueamento levou a escolha do classificador A1DE no cenário com discretização.



	Acc	F1	AUC	RMSE
BN	0,93	0,94	0,96	0,24
NB	0,93	0,94	0,97	0,24
A1DE	0,94	0,95	0,97	0,21
IBk -K 5 -A EuclideanDistance	0,93	0,94	0,97	0,26
J48 -U -M 1	0,91	0,93	0,95	0,29
RF -I 100 -k 0 -depth 0	0,93	0,94	0,96	0,25
SMO -C 1 -N 0 -K "NormalizedPolyKernel -E 1"	0,93	0,94	0,94	0,25



	Acc	F1	AUC	RMSE
LR-NB	0,85	0,87	0,93	0,34
NB	0,85	0,87	0,93	0,35
IBk -K 14 -F -A EuclideanDistance	0,91	0,93	0,94	0,33
J48 -B -C 0,76 -M 4	0,91	0,93	0,91	0,27
RF -I 80 -K 0 -depth 0	0,93	0,94	0,95	0,28
SMO -C 0,58 -N 1 -K "RBFKernel -G 9,63E-4"	0,9	0,92	0,93	0,3

Figura 30: Desempenho estimado para o diagnóstico de CCL usando *leave-one-out* para os cenários com e sem discretização (testes finais).

Tabela 36: Ranqueamento dos classificadores para o diagnóstico de CCL no cenário com discretização (testes finais).

Medida (pontuação)	BN	NB	A1DE	IBk -k 5 distância Euclidiana	J48 -U -M 1	RF -I 100 -k 0 -depth 0	SMO -C 1 -N 0 -K "NormalizedPoly Kernel -E 1"
Acc	0,93 (4)	0,93 (4)	0,94 (1)	0,93 (4)	0,91 (7)	0,93 (4)	0,93 (4)
F1	0,94 (4)	0,94 (4)	0,95 (1)	0,94 (4)	0,93 (7)	0,94 (4)	0,94 (4)
AUC	0,96 (4,5)	0,97 (2)	0,97 (2)	0,97 (2)	0,95 (6)	0,96 (4,5)	0,94 (7)
RMSE	0,24 (2,5)	0,24 (2,5)	0,21 (1)	0,26 (6)	0,29 (7)	0,25 (4,5)	0,25 (4,5)
Pontuação total	15	12,5	5	16	27	17	19,5
Posição	3	2	**1**	4	7	5	6

Tabela 37: Ranqueamento dos classificadores para o diagnóstico de CCL no cenário sem discretização (testes finais).

Medida (pontuação)	LR-NB	NB	IBk -K 14 -F distância Euclidiana	J48 -B -C 0,76 -M 4	RF -I 80 -K 0 -depth 0	SMO -C 0,58 -N 1 -K "RBFKernel -G 9,63E-4"
Acc	0,85 (5,5)	0,85 (5,5)	0,91 (2,5)	0,91 (2,5)	0,93 (1)	0,9 (4)
F1	0,87 (5,5)	0,87 (5,5)	0,93 (2,5)	0,93 (2,5)	0,94 (1)	0,92 (4)
AUC	0,93 (4)	0,93 (4)	0,94 (2)	0,91 (6)	0,95 (1)	0,93 (4)
RMSE	0,34 (5)	0,35 (6)	0,33 (4)	0,27 (1)	0,28 (2)	0,3 (3)
Pontuação total	20	21	11	12	5	15
Posição	5	6	2	3	**1**	4

Tabela 38: Escolha do melhor classificador/cenário para diagnóstico de CCL (testes finais).

Medida (pontuação)	com discretização A1DE	sem discretização RF -I 80 -K 0 -depth 0
Acc	0,94 (1)	0,93 (2)
F1	0,95 (1)	0,94 (2)
AUC	0,97 (1)	0,95 (2)
RMSE	0,21 (1)	0,28 (2)
Pontuação total	4	8
Posição	**1**	2

Note que, para Demência e CCL, a discretização supervisionada dos atributos contínuos contribuiu para a melhoria do desempenho geral dos classificadores (ver Figuras 28 e 30). Embora a sobreamostragem sobre o conjunto de dados desbalanceado de DA tenha contribuído para melhoria do desempenho geral quando a discretização não foi aplicada (ver

Figura 29), o cenário escolhido foi o com discretização e sem sobreamostragem para o classificador SVM que alcançou o melhor desempenho quando comparado aos outros classificadores avaliados (ver Tabelas 31 e 35).

6.3.2.2 ATRIBUTOS DOS CLASSIFICADORES BINÁRIOS OBTIDOS

As Tabelas 39-41 mostram os atributos selecionados para os classificadores binários obtidos respectivamente para o diagnóstico de Demência, DA e CCL após o processo de aprendizado automático utilizando dados combinados do CDA e do CRASI para treinamento. Tais tabelas também mostram os intervalos de discretização obtidos para os atributos contínuos que foram discretizados. Comparando-se a Tabela 22 com a Tabela 39 e a Tabela 22 com a Tabela 40, constata-se que a dimensionalidade (incluindo a classe) para Demência e DA foi reduzida de 15 para 14, porque o exame Stroop foi automaticamente excluído e nenhum exame exclusivo do CRASI foi incluído. Isto aconteceu porque o exame Stroop e os exames exclusivos do CRASI não atenderam à taxa máxima de dados não informados (70%) depois da adição dos dados do CRASI aos dados do CDA no sistema para Demência e DA. Comparando-se a Tabela 24 com a Tabela 41, constata-se que, para CCL, a dimensionalidade se manteve em 14 por causa da exclusão dos exames Stroop, IQCODE, Berg e Pfeffer (por causa da alta taxa de dados não informados) e da inclusão dos seguintes exames exclusivos do CRASI: Katz, CERAD memória de lista de palavras, TMT - Parte A - 1 OK e TCA-O que alcançaram percentual de dados não informados menor que 70% e ganho de informação não desprezível. Os exames TCA-E, TCA-TR e TCA-VAR, exclusivos do CRASI, não foram incluídos para o diagnóstico de CCL por apresentarem ganho de informação desprezível**.**

Tabela 39: Lista de atributos selecionados para diagnóstico de Demência (testes finais).

Atributo	**Valores possíveis**[21]
Idade	1:[0-76]; 2:[77-inf]
Gênero	1:feminino; 2:masculino
Escolaridade	1:[0-1]; 2:[2]; 3:[3-11]; 4:[12-inf)
MMSE	1:[0-12]; 2:[13-22];3:[23-26]; 4:[27-30]
CDR	1:0; 2:0,5; 3:1; 4:2; 5:3
VFT	1:[0-4]; 2:[5-11]; 3:[12-inf]
TMT – Parte A	1:[0-16]; 2:[17-59]; 3:[60-inf]
Lawton	1:[9-25]; 2:[26];3:[27]
CDT	1:0; 2:1; 3:2; 4:3; 5:4; 6:5
Pfeffer	1:0; 2:[1-2]; 3:[3-30]
IQCODE	1:[1-3,35]; 2:[3,36–5]

[21]Representa os valores discretos que o atributo pode assumir. A sintaxe para atributos que foram discretizados é n*:[J–K]*, onde n=valor, *J*=limite inferior do intervalo; *K*=limite superior do intervalo; *inf*=infinito. A discretização supervisionada estimou tais limites.

Berg	1:[0-51]; 2:[52-56]
Depressão	1:ausência; 2:presença

Tabela 40: Lista de atributos selecionados para diagnóstico de DA (testes finais).

Atributo	**Valores possíveis**
Idade	1:[0-70]; 2:[71-76]; 3:[77-81]; 4:[82:inf]
Gênero	1:feminino; 2:masculino
Escolaridade	1:[0-3]; 2:[4-5]; 3:[6-7]; 4:[8-inf]
MMSE	1:[0-10]; 2:[11-16];3:[17-20]; 4:[21-30]
CDR	1:0; 2:0,5; 3:1; 4:2; 5:3
VFT	1:[0-4]; 2:[5-7]; 3:[8-10]; 4:[11-inf]
TMT- Parte A	1:[0-22]; 2:[23-89]; 3:[90-140]; 4:[141-inf]
Lawton	1:[9-14]; 2:[15-19]; 3:[20-22]; 4:[23-27]
CDT	1:0; 2:1; 3:2; 4:3; 5:4; 6:5
Pfeffer	1:[0-11]; 2:[12-20]; 3:[21-27]; 4:[28-30]
IQCODE	1:[1-3,46]; 2:[3,47-3,97]; 3:[3,98-4,41]; 4:[4,42-5]
Berg	1:[0-40]; 2:[41-46]; 3:[47-51]; 4:[52-56]
Depressão	1:ausência; 2:presença

Tabela 41: Lista de atributos selecionados para diagnóstico de CCL (testes finais).

Atributo	**Valores possíveis**
Gênero	1:feminino; 2:masculino
Escolaridade	1:[0-15]; 2:[16-inf]
MMSE	1:[0-28]; 2:[29-30]
CDR	1:0; 2:0,5; 3:1; 4:2; 5:3
VFT	1:[0-17]; 2:[18-inf]
TMT – Parte A	1:[0-36]; 2:[37-inf]
Lawton	1:[9-24]; 2:[25-27]
CDT	1:0; 2:1; 3:2; 4:3; 5:4; 6:5
Depressão	1:ausência; 2:presença
Katz	1:0; 2:1; 3:2; 4:3; 5:4; 6:5; 7:6
CERAD memória de lista de palavras	1:[0-18]; 2:[19-30]
TMT-Parte A-1 OK	1:0;2:1
TCA-O	1:[0-13]; 2:[14-100]

6.3.3 TESTES COM DADOS NÃO UTILIZADOS NO TREINAMENTO

As Tabelas 42-44 mostram as matrizes confusão obtidas pelos testes usando os dados do CRASI (ver Figura 27 (d-f)) para os classificadores automaticamente treinados para Demência, DA e CCL (respectivamente RF -I 7 -K 0 -depth 4, SMO -C 0,77 -N 2 -K "PolyKernel -E 1,26 -L" e A1DE). Para a realização dos testes, exames presentes na base de testes que não estavam presentes nos classificadores obtidos não foram considerados. Exames presentes apenas nos classificadores obtidos, mas não na base de testes, foram considerados dados não informados.

Para Demência, o classificador acertou a classificação de todas as tuplas de teste. Portanto, o desempenho para Demência foi melhor que o estimado pelo *leave-one-out*: Acc de 92%, F1 de 0,94, AUC de 0,95 e RMSE de 0,26 (ver Tabela 30).

Para DA, apenas 11 tuplas foram testadas (ver Figura 27 (e)). O classificador errou a classificação para as tuplas de teste negativas classificando-as erroneamente como positivas.

Para CCL, os resultados para base de teste separada foram: Acc de 98 %, F1 de 0,99, AUC de 0,73 e RMSE de 0,13. Portanto, o desempenho para CCL foi muito bom quando comparado ao estimado pelo *leave-one-out*: Acc de 94 %, F1 de 0,95, AUC de 0,97 e RMSE de 0,21 (ver Tabela 38).

Tabela 42: Matriz confusão obtida para Demência (testes finais).

Predito	***Ground Truth***	
	Negativo	**Positivo**
Negativo	98	0
Positivo	0	26

Tabela 43: Matriz confusão obtida para DA (testes finais).

Predito	***Ground Truth***	
	Negativo	**Positivo**
Negativo	0	0
Positivo	5	6

Tabela 44: Matriz confusão obtida para CCL (testes finais).

Predito	***Ground Truth***	
	Negativo	**Positivo**
Negativo	11	1
Positivo	1	85

6.4 RELEVÂNCIA DOS EXAMES

A Tabela 45 mostra os três exames com maiores valores de FC médio dentre os exames presentes nos classificadores obtidos para Demência, DA e CCL pelo processo automático de aprendizado. Os valores de FC médio foram calculados aplicando-se a metodologia descrita na Seção 4.2.2 considerando que nenhum dado a priori sobre o paciente foi informado. Portanto, quanto maior o valor do FC médio, mais relevante é considerado o exame.

Para fins comparativos, a Tabela 46 mostra os três exames mais relevantes se o ranqueamento pelo valor do ganho de informação fosse aplicado. Este não é o caso, pois a

metodologia do FC médio é a que vale para classificadores treinados em cenários com discretização (caso dos classificadores finais obtidos para Demência, DA e CCL).

Visando avaliar a relevância dos exames apontada pelo modelo de decisão obtido para o sistema, dois médicos especialistas, um médico do CDA e uma médica do CRASI, avaliaram cada exame que costumam aplicar em seus pacientes em termos de sua relevância para o diagnóstico de Demência, DA e CCL de acordo com três níveis de relevância: (+) pouco relevante, (++) relevante ou (+++) muito relevante. As indicações dos especialistas estão na Tabela 47. Exames exclusivos do CRASI foram avaliados pela médica do CRASI e os demais exames foram avaliados pelo médico do CDA.

Comparando-se as Tabelas 45 e 46 com a Tabela 47, constata-se que a metodologia do FC médio alcançou melhor concordância com a opinião dos especialistas do que o ranqueamento pelo valor do ganho de informação. Quanto aos exames mais relevantes apontados pela metodologia do FC médio (Tabela 45), a médica especialista do CRASI ainda ponderou que tais exames são realmente importantes para o diagnóstico diferencial. Ela ressaltou que o sistema não apontou nenhum exame de domínio exclusivo como mais relevante (por exemplo, o teste de trilhas, que só pode ser realizado por psicólogos) e que os exames apontados pertencem a domínios múltiplos e, como tal, podem ser aplicados por médicos de diferentes especialidades na triagem de pacientes, o que considerou vantajoso.

Esta seção evidencia que a metodologia proposta para a recomendação de exames foi uma boa escolha.

Tabela 45: Exames mais relevantes para o diagnóstico de Demência, DA e CCL de acordo com os classificadores obtidos.

Doença/Transtorno	Exame	FC médio
Demência	CDR	0,72
	MMSE	0,71
	VFT	0,64
DA	CDR	0,99
	Lawton	0,99
	Berg	0,99
CCL	CDR	0,92
	Pfeffer	0,83
	CERAD memória de lista de palavras	0,83

Tabela 46: Exames mais relevantes para o diagnóstico de Demência, DA e CCL de acordo com o ganho de informação.

Doença/Transtorno	Exame	Ganho de informação
Demência	CDR	0,49
	MMSE	0,38
	VFT	0,26
DA	MMSE	0,039
	CDT	0,031
	Pfeffer	0,020
CCL	CDR	0,34
	MMSE	0,20
	CDT	0,13

Tabela 47: Relevância dos exames para diagnóstico de Demência, DA e CCL de acordo com médicos especialistas[a].

Exame	Demência	DA	CCL
CDR	+++	+++	+++
Pfeffer	+++	+++	+++
Lawton	+++	+++	+++
MMSE	+++	+++	++
VFT	+++	+	++
Depressão	++	++	++
TMT – Parte A	++	++	++
Stroop	++	++	++
Berg	++	++	++
IQCODE	++	++	++
CDT	++	+	++
Katz	++	++	++
TMT - Parte A - 1 OK	++	++	++
CERAD memória de lista de palavras	++	++	++
TCA-O	++	++	++
TCA-E	++	++	++
TCA-TR	+++	+++	+++
TCA-VAR	+++	+++	+++

[a] Níveis de relevância: (+) pouco relevante, (++) relevante ou (+++) muito relevante; marcação azul foi utilizada em exames listados na Tabela 45 e marcação cinza foi utilizada em exames listados na Tabela 46.

6.5 DISCUSSÃO

Como visto anteriormente, a implementação do modelo de decisão dinâmico proposto nesta tese foi capaz de incluir novos exames para o diagnóstico de CCL quando os dados do CRASI foram adicionados ao sistema. Isto aconteceu porque dados de uma quantidade relativamente grande de pacientes foi adicionada para CCL. Dados de 60 pacientes contendo pontuações para novos exames foram adicionados a dados de 67 pacientes que já existiam no

banco de dados e que foram coletados do CDA (ver Figura 27 (c)). A inclusão de novos exames para o diagnóstico de Demência e DA acontecerá automaticamente na medida em que o sistema for utilizado na prática, porque o modelo de decisão dinâmico é capaz de incluir exames frequentemente aplicados aos pacientes nos centros de saúde, utilizando o CDSS, desde que tais exames sejam suficientemente informados e apresentem ganho de informação não desprezível.

Além disso, o mecanismo de recomendação do CDSS quanto aos exames mais relevantes a serem aplicados apresentou resultados que concordaram apropriadamente com a opinião de médicos especialistas, sendo uma ferramenta útil para ajudar médicos menos especializados ou pouco experientes na priorização dos exames a serem aplicados. Uma limitação da presente tese reside na metodologia da Seção 4.2.1 (indicação de dados de saúde informados mais relevantes para um dado diagnóstico sugerido) não ter sido avaliada. Outra limitação é que esta indicação, para cenários com discretização, baseia-se no cálculo do FC por atributo individualmente atribuído, contudo, a saída do classificador pode ser bastante alterada de acordo com a combinação dos atributos informados. Por isso, há a necessidade de se estudar outras metodologias para a indicação dos exames informados mais relevantes para um diagnóstico sugerido.

Na comparação entre os desempenhos obtidos pela aplicação do modelo de decisão dinâmico proposto ao CDSS de apoio ao diagnóstico de Demência, DA e CCL e os desempenhos obtidos por outros trabalhos na predição diagnóstica de DA e transtornos relacionados baseada em modelos dinâmicos (CHAN et al., 1993; HUANG e CHEN, 2015; CHO et al., 2012; LEE et al., 2014), deve-se notar que existem diferenças em termos de tipos de atributos, métodos para avaliação dos classificadores e rótulos de classe adotados.

O desempenho estimado final do classificador obtido para o diagnóstico de Demência (diagnóstico quanto a um paciente possuir ou não Demência sem distinguir se Demência é do tipo DA ou não) usando o método de avaliação *leave-one- out* foi: Acc de 92%, F1 de 0,94, AUC de 0,95 e RMSE de 0,26.

O desempenho estimado do classificador obtido para o diagnóstico de DA, que distingue, para pacientes dementes, se a Demência é devida à DA ou não, foi similar ao alcançado por trabalhos anteriores em termos de Acc: a Acc do classificador obtido para diagnóstico de DA, usando *leave-one-out*, foi 85%, enquanto CHAN et al. (1993), CHO et al. (2012) e LEE et al. (2014) alcançaram respectivamente Acc de 85%, 86% e 87,5%. Entretanto, as acurácias reportadas por tais trabalhos referem-se à classificação de indivíduos com DA versus indivíduos normais. HUANG e CHEN (2015) reportaram Acc de 88,5%

considerando o problema de classificação com três classes: pacientes com DA, com CCL e controles (normais) e, portanto, tal referência não pode ser usada na comparação com a classificação binária DA positivo versus DA negativo.

O desempenho estimado do classificador obtido para o diagnóstico de CCL, que distingue pacientes com CCL de pacientes normais, alcançou Acc de 94% usando *leave-one-out*, portanto, superando CHO et al. (2012) (que apresentou Acc de 82% na predição paciente CCL-c versus N), mas os rótulos possuem significados diferentes.

Quanto ao teste da generalização (dados de teste em separado) do classificador de Demência, a generalização obtida foi surpreendente, pois, nos testes finais, o classificador predisse corretamente a classe para todas as tuplas de teste, conforme apresentado na Tabela 42. Há a necessidade de se estudar as razões desta generalização surpreendente, pois os dados para Demência podem, por exemplo, seguir padrões muito simples não requerendo aprendizado de máquina. Outra possibilidade é as tuplas de teste para Demência estarem com um perfil muito semelhante entre os diversos pacientes, de forma que a base de testes de Demência pode não estar suficientemente abrangente para englobar perfis mais distintos. Como a generalização para Demência também foi muito boa para os testes iniciais (ver Tabela 25), em que a base de treinamento era do CDA e a base de teste era do CRASI, outra possibilidade é que um ou mais atributos em comum das bases do CDA e do CRASI sejam muito mais importantes do que os outros para o diagnóstico de Demência.

Para DA, os testes com dados em separado utilizaram um número muito pequeno de tuplas (testes iniciais: 4 tuplas e testes finais: 11 tuplas). Para superar este problema, uma possibilidade é dividir as bases de treinamento e teste de outra maneira, não seguindo a ordem cronológica adotada por esta tese. Outra possibilidade é coletar mais dados e/ou identificar novas bases com diagnóstico diferencial entre DA e demências de outros tipos. De acordo com as Tabelas 26 e 43, os classificadores obtidos para diagnóstico de DA classificaram as tuplas de teste sempre como positivas. É necessário investigar, se esta limitação persistirá no caso de se aumentar o número de tuplas de teste para DA. Caso isto continue acontecendo, pode ser um indicativo de que o problema de desbalanceamento da base não esteja sendo tratado a contento pelo processo de aprendizado da Figura 12. Neste caso, uma possível solução é alterar este processo para que, no caso de bases desbalanceadas, sejam criados cenários com diferentes taxas de sobreamostragem de acordo com o nível de desbalanceamento da base, ou seja, criar cenários aplicando taxas de sobreamostragem maiores para bases mais desbalanceadas. Atualmente, a taxa de sobreamostragem está fixada em 100%.

Para CCL, os testes com dados em separado evidenciaram que a generalização do classificador para diagnóstico de CCL melhorou quando os dados do CRASI foram adicionados aos dados do CDA para treinamento, pois houve redução do número de falsos negativos (ver Tabelas 27 e 44).

CAPÍTULO 7 – CONCLUSÃO

A maioria dos trabalhos, que propõe o diagnóstico auxiliado por computador empregando aprendizado de máquina para diagnosticar doenças/transtornos, está baseada em modelos estáticos, isto é, modelos construídos utilizando bases de treinamento que não sofrem atualizações. Nesses trabalhos relacionados, não é possível aproveitar a realimentação diagnóstica que pode ser provida pelo médico ao CDSS, durante a rotina clínica, para melhorar o modelo de decisão. Adicionalmente, foram pesquisados trabalhos de auxílio a diagnóstico baseados em modelos dinâmicos, contudo, nenhum deles apresentava a característica necessária de ser adequado a diferentes centros de saúde utilizando diferentes conjuntos de exames para subsidiar a decisão diagnóstica.

Visando suprir tal necessidade, a presente tese propôs um modelo de decisão dinâmico não somente às atualizações da base de treinamento, mas a critérios de diagnóstico dinâmicos, sendo capaz de incluir atributos conforme estes se tornem potencialmente importantes para o diagnóstico. Para isto, um processo automático de aprendizado periodicamente verifica se é necessário reconstruir o modelo para as doenças/transtornos cujos diagnósticos se deseja auxiliar e, caso seja necessário, tal processo retreina o classificador respectivo. O modelo de decisão dinâmico proposto baseia-se em uma metodologia genérica que pode ser utilizada por sistemas de apoio ao diagnóstico de diversas doenças/transtornos. Contudo, é importante ressaltar que o modelo de decisão dinâmico proposto é restrito a sistemas em que os tipos de dados usados na predição do diagnóstico possam ser representados por atributos contínuos ou discretos. Por exemplo, o modelo proposto pode ser aplicado ao diagnóstico de doenças que requerem exames laboratoriais, como exames de sangue.

Diante do crescimento da prevalência da Demência, da DA e do CCL, causado pelo fenômeno do envelhecimento global, este trabalho aplicou o modelo de decisão dinâmico proposto a um CDSS projetado para auxiliar profissionais de saúde na tomada de decisão durante o processo diagnóstico com base no registro de evidências clínicas incluindo as pontuações obtidas em TNPs aplicados sobre os pacientes.

A flexibilidade do CDSS desenvolvido permite ao médico definir no sistema um novo tipo de exame (TNP) e, a partir de então, as pontuações obtidas na aplicação deste novo TNP aos pacientes podem ser armazenadas no sistema. O processo de aprendizado automaticamente incluirá este novo TNP como um novo atributo no modelo decisão quando tal TNP tiver suas pontuações informadas com frequência nas decisões diagnósticas tomadas

e contribuir para separar o diagnóstico positivo do negativo (isto é, atender a um percentual mínimo de dados informados e apresentar ganho de informação não desprezível).

O processo de aprendizado do modelo de decisão dinâmico proposto inclui etapas de pré-processamento que não são fixas, isto é, são gerados diversos cenários de pré-processamento incluindo a aplicação ou não de discretização supervisionada sobre os atributos contínuos e, no caso de bases desbalanceadas, a aplicação ou não de sobreamostragem. A criação de tais cenários visa avaliar se a discretização supervisionada e a sobreamostragem podem contribuir para a melhoria do desempenho na classificação. Além disso, nem mesmo o algoritmo de classificação é fixo, mas um conjunto de algoritmos de classificação é avaliado usando CV e as bases de treinamento geradas de acordo com os cenários de pré-processamento. Alguns classificadores avaliados têm seus hiperparâmetros ajustados automaticamente e, após a avaliação, que considera diversas medidas de desempenho, é realizado o ranqueamento dos classificadores de acordo com os valores de desempenho por eles obtidos para que seja escolhido o classificador (e cenário associado) com melhor desempenho, considerando o conjunto inteiro de medidas. Por fim, o classificador escolhido é treinado de acordo com o cenário associado e o modelo obtido é armazenado.

O CDSS flexível desenvolvido para implementar o modelo de decisão dinâmico possui a vantagem de poder ser facilmente acessado a partir de dispositivos móveis desde que possuam um navegador web com JS habilitado. O acesso móvel a sistemas de apoio ao diagnóstico de transtornos que envolvam o comprometimento cognitivo é importante, pois, com frequência, os pacientes sendo avaliados possuem dificuldades de locomoção, e neste caso, o atendimento médico pode ocorrer em lugares mais convenientes aos pacientes. O CDSS desenvolvido possui uma arquitetura que inclui componentes abertos e reutilizáveis, que podem servir de base para outros sistemas de apoio diagnóstico provendo bases para futuros avanços nos cuidados à saúde.

Testes experimentais foram realizados para avaliar os resultados da implementação da metodologia proposta e a capacidade de generalização dos classificadores obtidos para diagnosticar Demência, DA e CCL. Foram utilizadas bases de dados reais de dois centros de saúde (CDA/UFRJ e CRASI/UFF), as quais contêm conjuntos distintos, porém não disjuntos, de exames mais frequentemente aplicados no diagnóstico de Demência, DA e CCL. O processo automático de aprendizado aplicado às respectivas bases de treinamento escolheu cenários com discretização supervisionada de atributos contínuos. Isto evidenciou que tal discretização foi benéfica para as bases. O classificador obtido para diagnóstico de CCL nos

testes experimentais finais mostra que o modelo de decisão incluiu exames exclusivos do CRASI, quando os dados do CRASI foram incluídos no treinamento, sendo, agora, atributos do classificador que subsidia o diagnóstico de CCL. Além disto, para CCL, este classificador final obtido melhorou a sua generalização quando comparado ao classificador obtido nos testes experimentais iniciais.

Quanto aos TNPs considerados mais importantes de acordo com a recomendação de exames futuros, implementada no CDSS, estes aparecem na avaliação dos especialistas como muito relevantes ou relevantes, o que evidencia a robustez da metodologia proposta para a recomendação de exames.

Como dito anteriormente, na comparação entre os desempenhos dos classificadores obtidos na presente tese e os desempenhos reportados por outros trabalhos que também utilizam modelos de decisão dinâmicos na predição diagnóstica de DA e de transtornos relacionados, existem diferenças em termos de métodos de avaliação, rótulos de classe e tipos de atributos. Quanto aos tipos de atributos, por exemplo, a predição diagnóstica do CDSS desenvolvido baseia-se em fatores predisponentes e pontuações obtidas em TNPs, enquanto tais trabalhos relacionados envolvem análise de imagens. Realizando-se a comparação mesmo assim, constata-se que os resultados de desempenho obtidos pelo modelo de decisão dinâmico proposto são competitivos com a vantagem de permitir a inclusão de atributos sendo, portanto, mais flexível e apto à utilização por centros de saúde distintos. O modelo de suporte à decisão proposto alcançou bom desempenho estimado nos testes finais, sobretudo para Demência (acurácia de 92%) e CCL (acurácia de 94%).

7.1 PRINCIPAIS CONTRIBUIÇÕES

Ressalta-se que as contribuições desta tese de doutorado foram:

1. Proposição de um modelo genérico de suporte à decisão diagnóstica, o qual é dinâmico e flexível, isto é, ajusta-se de acordo com novos dados contendo diagnósticos providos por médicos especialistas levando em consideração que diferentes centros de saúde adotam diferentes conjuntos de exames para diagnosticar e que critérios de diagnóstico evoluem com o tempo. O dinamismo do modelo de decisão proposto é realizado através de um método de aprendizado que automaticamente o retreina com a inclusão de exames aplicados mais frequentemente para diagnosticar e potencialmente relevantes, na medida em que os dados dos pacientes são atualizados por especialistas destes centros de saúde.

2. Projeto e desenvolvimento de um CDSS que implementa o modelo de decisão dinâmico proposto para apoiar o diagnóstico de Demência, DA e CCL auxiliando profissionais de saúde com uma segunda opinião e visando à diminuição dos erros diagnósticos.
    a. O modelo de decisão do sistema é composto por três classificadores binários para predizer o diagnóstico mais provável (negativo ou positivo) para Demência, DA e CCL baseando-se em fatores predisponentes e pontuações obtidas para TNPs aplicados aos pacientes.
    b. O sistema tem potencial de ser largamente utilizado por ser acessado através de dispositivos móveis e foi projetado e desenvolvido para ter uma arquitetura baseada em padrões abertos e bem documentados, refletindo boas práticas de codificação orientada a objetos, o que permite a reutilização de componentes por outros sistemas de apoio diagnóstico.
3. A implementação de dois algoritmos de classificação que são automaticamente avaliados e podem ser escolhidos pelo método de aprendizado do modelo proposto:
    a. Uma RB discreta com estrutura fornecida, composta por três níveis, automatizando a modelagem e o aprendizado de parâmetros propostos por SEIXAS et al. (2014).
    b. Um modelo híbrido combinando Regressão Logística e Naïve Bayes, como descrito por CARVALHO et al. (2017b), equivalente a uma RB com estrutura fornecida composta por três níveis, porém capaz de combinar nós discretos e contínuos.

7.2 TRABALHOS FUTUROS

Um trabalho a ser realizado consiste em implantar o CDSS desenvolvido na rotina clínica do CDA e do CRASI, obtendo *feedback* dos especialistas quanto à efetividade do sistema na redução dos erros diagnósticos e identificando possíveis oportunidades de melhoria do sistema. Como ferramenta de avaliação da usabilidade do sistema, foi incorporada ao CDSS uma versão eletrônica de um questionário baseado na escala SUS - *System Usability Scale* (BANGOR et al., 2008), contendo dez perguntas com cinco alternativas que proporciona uma visão do ponto de vista do usuário quanto à facilidade de usar o sistema resultando em uma pontuação de 0 a 100. Oportunidades de melhoria do

sistema podem ser identificadas pelo preenchimento deste questionário pelos médicos que utilizarão o sistema.

De forma a incentivar a adoção em larga escala do sistema, outro trabalho futuro é a automatização computacional de alguns TNPs propícios a isso como o Stroop e o TCA integrando a realização destes testes ao CDSS.

Os testes experimentais realizados por esta tese também podem ser melhorados. Para isto, uma sugestão de trabalho futuro consiste em realizar mais testes experimentais alterando a ordem de adição dos dados na construção do modelo. Primeiramente, pode-se utilizar apenas dados do CRASI para testar o processo de aprendizado e os dados do CDA para testar a generalização dos classificadores obtidos. Posteriormente, pode-se adicionar, à base do CRASI, parte da base do CDA para testar o processo de aprendizado e testar a generalização dos classificadores obtidos com a outra parte da base do CDA.

Outra possibilidade de trabalho futuro é misturar aleatoriamente todos os dados (CDA e CRASI), separar uma parte para teste do processo de aprendizado e a outra parte para teste da generalização dos modelos obtidos, mantendo as mesmas proporções das classes. Se adotada, esta sugestão permitirá aumentar o número de instâncias para teste da generalização do classificador para diagnóstico de DA.

Outro trabalho futuro, como mencionado na Seção 6.5, é estudar as razões do desempenho de 100% nos testes finais realizados para testar a generalização para Demência.

Além disso, como mencionado na Seção 6.5, o processo de aprendizado pode ser modificado para que, no caso de bases desbalanceadas, sejam criados cenários com diferentes taxas de sobreamostragem de acordo com o nível de desbalanceamento da base, pois, atualmente a taxa de sobreamostragem está fixada em 100%. Com isto, espera-se resolver a limitação de que os classificadores obtidos para DA (nos testes com dados em separado realizados até agora) apontaram para a classe positiva para todas as tuplas de teste.

O ranqueamento realizado pelo processo de aprendizado descrito nesta tese atribui igual importância para todas as medidas de desempenho. Um trabalho futuro é realizar uma análise de sensibilidade para verificar a equivalência das medidas de desempenho para as bases utilizadas. Além disso, pode-se atribuir pesos diferentes no processo de ranqueamento para as medidas de desempenho consideradas não equivalentes, de acordo com a importância de tais medidas para um dado diagnóstico.

Atualmente, o processo de aprendizado verifica semanalmente, para cada doença/transtorno, se existe ao menos um diagnóstico novo inserido pelo médico especialista e, caso exista, o modelo é automaticamente reconstruído. Um trabalho futuro é estudar

alternativas que modifiquem a condição de disparo deste retreinamento. Por exemplo, o modelo poderia ser automaticamente retreinado a cada número x de diagnósticos novos inseridos no sistema pelo médico especialista e a determinação deste x seria objeto de estudo. Outra possibilidade é retreinar o modelo a cada y diagnósticos sugeridos pelo sistema com os quais o médico especialista discorde, ou seja, a cada y erros de diagnóstico do modelo. Neste caso, a determinação desta quantidade y também seria objeto de estudo.

Outro trabalho futuro é investigar a escalabilidade do processo de aprendizado do modelo de decisão dinâmico proposto, pois tal processo é baseado em retreinamento. Pretende-se estimar a taxa de diagnósticos a serem realizados nos próximos anos por médicos do CDA e do CRASI que potencialmente utilizarão o sistema e realizar, com base nesta estimativa, testes de escalabilidade com a criação de dados artificiais monitorando-se os tempos de execução e possíveis problemas de implementação que venham a surgir.

Como descrito anteriormente, para os dados atuais utilizados nos testes experimentais finais, o processo de aprendizado levou cerca de duas horas e meia para ser executado sendo que 90% deste tempo foi gasto com o ajuste dos hiperparâmetros, que é um processo computacionalmente custoso. Portanto, a avaliação (usando *leave-one-out*) dos diversos classificadores com hiperparâmetros já ajustados considerando os diversos cenários de pré-processamento levou menos que os 15 minutos restantes (10% do tempo total de execução) para ser executada. O método de avaliação *leave-one-out* é indicado para bases pequenas por consumir mais tempo que outros métodos de avaliação. Com o uso do sistema, as bases de treinamento crescerão, de forma que o tempo dispendido na avaliação dos classificadores tende a crescer. Neste caso, uma maneira de reduzir o tempo de execução total do processo de aprendizado é substituir o *leave-one-out* por 10-*fold* CV na forma estratificada e realizar testes estatísticos para comparar o desempenho entre os classificadores. Em testes realizados usando as bases atuais que são pequenas, 10-*fold* CV na forma estratificada com teste estatístico *t-Student* resultou em muitos empates dificultando a escolha do melhor classificador e cenário e, portanto, não foi adotada, mas com o crescimento das bases, espera-se que isto não mais ocorra, se os desvios padrões nos valores obtidos para as medidas de desempenho se reduzirem.

Outro trabalho futuro, considerando as bases pequenas como estão no momento, é modificar o processo de aprendizado para manter o *leave-one-out* como método para a avaliação dos classificadores, mas incluir o teste estatístico de *Friedman*. Este teste usualmente é aplicado para comparar o desempenho de múltiplos classificadores considerando múltiplas bases de treinamento, porém, no caso do processo de aprendizado da

tese, pode ser utilizado para comparar o desempenho de múltiplos classificadores sobre a mesma base de treinamento, mas considerando múltiplas medidas de desempenho.

Na investigação da escalabilidade, pode-se adotar a alternativa anteriormente mencionada de mudar o método de avaliação dos classificadores para 10-*fold* CV na forma estratificada com teste estatístico *t-Student*. Porém, caso os resultados alcançados para o desempenho computacional não sejam considerados razoáveis, pode-se considerar o uso de paralelização e GPUs. Outra possibilidade é a reformulação do modelo de decisão dinâmico genérico proposto e de seu processo de aprendizado associado. Uma proposta para esta reformulação é a criação de vários modelos capazes de lidar com observações parciais: um modelo para cada hospital contendo atributos exclusivos de cada um deles, um modelo contendo atributos em comum e outro modelo contendo novos atributos não contidos nos modelos anteriores. Na fase de treinamento pré-implantação do sistema, dados rotulados comporão os modelos de cada hospital e o modelo com atributos em comum de acordo com os atributos informados. Então, os modelos serão combinados usando a abordagem *stacking*, isto é, as saídas (predições) dos modelos construídos serão consideradas entradas (atributos) para um classificador no último nível sendo responsável pela agregação das predições para compor a predição final. Este classificador de último nível também deverá ser capaz de lidar com observações parciais, pois na fase pré-implantação do sistema, o modelo com novos atributos ainda não terá sido construído e a entrada correspondente no classificador de último nível para este modelo não será informada. Durante o uso do sistema, quando chegarem dados rotulados, o aprendizado será incremental apenas no classificador do último nível. Se novos atributos estiverem presentes nestes dados rotulados, um modelo contendo os novos atributos será criado e, a partir de então, suas saídas serão utilizadas como entrada na atualização do classificador do último nível. Durante a predição, os atributos informados determinarão quais saídas de quais modelos serão agregadas na determinação da classe pelo classificador do último nível.

Outro trabalho futuro que pode ser realizado é a alteração da implementação do modelo de decisão dinâmico incorporando o processo diagnóstico da Figura 15 a um classificador hierárquico único. Cabe ressaltar que, neste caso, o fluxo de exibição das sugestões diagnósticas (Figura 17) também seria alterado com a exibição do diagnóstico mais provável considerando todos os níveis da hierarquia. Neste caso, o modelo de decisão dinâmico genérico deverá ser modificado para permitir a avaliação de classificadores hierárquicos capazes de lidar com observações parciais. Ainda, é preciso avaliar como a incorporação do processo diagnóstico da Figura 15 a um classificador hierárquico único

afetaria a flexibilidade do modelo, principalmente para o caso da solução do parágrafo anterior ser adotada.

Como visto na Seção 6.5, a metodologia de indicação dos exames informados mais relevantes para um diagnóstico sugerido, para classificadores treinados em cenários com discretização, baseia-se no cálculo do FC por atributo individualmente atribuído. Contudo, a saída do classificador pode ser bastante alterada de acordo com diferentes combinações dos atributos informados para um dado paciente. Por causa desta limitação, pode-se utilizar o ranqueamento pelo ganho de informação, que já é utilizado para classificadores treinados nos cenários sem discretização ou adotar outras metodologias. Por exemplo, (RIBEIRO, SINGH E GUESTRIN, 2016) descrevem um método agnóstico, ou seja, independente do algoritmo de classificação, que pode ser empregado para indicar os exames informados mais relevantes para um diagnóstico sugerido.

A criação de modelos individualizados, calibrados de acordo com o paciente, também pode ser objeto de estudo futuro.

Outro trabalho futuro interessante é a utilização do modelo de decisão dinâmico proposto a outros tipos de doenças/transtornos, verificando a generalidade do modelo em outros processos de diagnóstico.

REFERÊNCIAS

ABU-MOSTAFA, Yaser S.; MAGDON-ISMAIL, Malik; LIN, Hsuan-Tien. **Learning from data**. New York, NY, USA: AMLBook, 2012.

AHA, D. W.; KIBLER, D.; ALBERT, M. K. Instance-based learning algorithms. **Machine Learning**, v. 6, n. 1, p. 37-66, 1991.

AHMAD, M.A.; KHAN, N.A.; MAJEED, W. Computer assisted analysis system of electroencephalogram for diagnosing epilepsy. Em: **Pattern Recognition (ICPR), 2014 22nd International Conference on**. IEEE, 2014. p. 3386-3391.

ALBERT, M. S; DEKOSKY, S.T.; DICKSON, D.; DUBOIS, B.; FELDMAN, H.H.; FOX, N.C.; GAMST, A.; HOLTZMAN, D.M.; JAGUST, W.J.; PETERSEN, R.C.; SNYDER, P.J.; CARRILO, M.C.; THIES,B.; PHELPS, C.H. The diagnosis of mild cognitive impairment due to Alzheimer's disease: Recommendations from the National Institute on Aging-Alzheimer's Association workgroups on diagnostic guidelines for Alzheimer's disease. **Alzheimer's & dementia**, v. 7, n. 3, p. 270-279, 2011.

ALZHEIMER'S A. 2017 Alzheimer's disease facts and figures. **Alzheimer's & Dementia**, v. 13, n. 4, p. 325-373, 2017.

ANWAR, T.; ASGHAR, S.; FONG, S. Bayesian based subgroup discovery. Em: **Digital Information Management (ICDIM), 2011 Sixth International Conference on**. IEEE, 2011. p. 154-161.

BANGOR, A.; KORTUM, P. T.; MILLER, J. T. An empirical evaluation of the system usability scale. **Intl Journal of Human–Computer Interaction**, v. 24, n. 6, p. 574-94, 2008.

BATTISTA, P.; SALVATORE, C.; CASTIGLIONI, I. Optimizing neuropsychological assessments for cognitive, behavioral, and functional impairment classification: a machine learning study. **Behavioural Neurology**, v. 2017, 2017.

BHATKOTI, P.; PAUL, M. Early diagnosis of Alzheimer's disease: A multi-class deep learning framework with modified k-sparse autoencoder classification. Em: **Image and Vision Computing New Zealand (IVCNZ), 2016 International Conference on**. IEEE, 2016. p. 1-5.

BERNER, E.S. **Clinical decision support systems: theory and practice**. Birminghain: Springer, 2007.

BERTOLUCCI, P. H. F.; OKAMOTO, I.H.; BRUCKI, S.M.D.; SIVIERO, M.O.; NETO, J.T.; RAMOS, L.R. Applicability of the CERAD neuropsychological battery to Brazilian elderly. **Arquivos de Neuro-psiquiatria**, v. 59, n. 3A, p. 532-536, 2001.

BOTTINO, C.M.c.; AZEVEDO, D.; TATSCH, M.; HOTOTIAN, S.R.; MOSCOSO, M.A.; FOLQUITO, J.; SCALCO, A.Z.; BAZZARELLA, M.C.; LOPES, M.A.; LITVOC, J. Estimate of dementia prevalence in a community sample from São Paulo, Brazil. **Dementia and Geriatric Cognitive Disorders**, v. 26, n. 4, p. 291-299, 2008.

BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5-32, 2001.

CARAMELLI, P.; BARBOSA, M.T.L. Como diagnosticar as quatro causas mais freqüentes de demência? How to diagnose the four most frequent causes of dementia? **Revista Brasileira de Psiquiatria**, v. 24, n. Supl I, p. 7-10, 2002.

CARVALHO, C. M.; MUCHALUAT-SAADE, D. C.; CONCI, A.; SEIXAS, F.L.; LAKS, J. A Clinical Decision Support System for Aiding Diagnosis of Alzheimer's Disease and Related

Disorders in Mobile Devices. Em: **IEEE International Conference on Communications (ICC)**, 2017.

CARVALHO, C. M.; SEIXAS, F.L.; MUCHALUAT-SAADE, D. C.; CONCI, A.; LAKS, J. Improving a Bayesian Decision Model for Supporting Diagnosis of Alzheimer's Disease and Related Disorders. Em: **13th International Conference on Machine Learning and Data Mining**, 2017.

CARVALHO, C. M.; SEIXAS, F.L.; MUCHALUAT-SAADE, D. C.; CONCI, A.; BOECHAT, Y., LAKS, J. A System for Aiding Diagnosis of Alzheimer's Disease and Related Disorders with an Adaptable Decision Model. Em: **International Joint Conference on Neural Networks (IJCNN)**, IEEE, 2018.

CHAN, M.; ANDRE, B.; HERRERA, A.; CELSIS, P. Incremental learning in a multilayer neural network as an aid to Alzheimer's disease diagnosis. Em: **Systems, Man and Cybernetics, 1993.'Systems Engineering in the Service of Humans', Conference Proceedings., International Conference on**. IEEE, 1993. p. 1-4.

CHAVES, R.; RAMÍREZ, J.; GORRIZ, J.; ADNI. Integrating discretization and association rule-based classification for Alzheimer's disease diagnosis. **Expert Systems with Applications**, v. 40, n. 5, p. 1571-1578, 2013.

CHO, Y.; SEONG, J.K.; JEONG, Y.; SHIN, S.Y.; ADNI. Individual subject classification for Alzheimer's disease based on incremental learning using a spatial frequency representation of cortical thickness data. **Neuroimage**, v. 59, n. 3, p. 2217-2230, 2012.

CELSIS, P. Age-related cognitive decline, mild cognitive impairment or preclinical Alzheimer's disease? **Annals of Medicine**, v. 32, n. 1, p. 6-14, 2000.

CHAWLA, N.; BOWYER, K.; HALL, L.; KEGELMEYER, W. Smote: synthetic minority over-sampling technique. **Journal of Artificial Intelligence Research**, v. 16, p. 321–357, 2002.

CHEN, C.; LIU, Z.; LI, H.; ZHOU, R.; ZHANG, Y.; LIU, R. EEG Detection Based on Wavelet Transform and SVM Method. Em: **Smart Cloud (SmartCloud), IEEE International Conference on**. IEEE, 2016. p. 241-247.

COHN, D. A.; GHAHRAMANI, Z.; JORDAN, M. I. Active learning with statistical models. **Journal of Artificial Intelligence Research**, 4, p. 129-145, 1996.

COOPER, G. F.; HERSKOVITS, E. A Bayesian method for the induction of probabilistic networks from data. **Machine Learning**, v. 9, n. 4, p. 309-347, 1992.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273-297, 1995.

COSTA, L.; GAGO, M.F.; YELSHYNA, D.; FERREIRA, J.; SILVA, H.D.; ROCHA, L.; SOUSA, N. BICHO, E. Application of machine learning in postural control kinematics for the diagnosis of Alzheimer's disease. **Computational Intelligence and Neuroscience**, v. 2016, 2016.

DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the EM algorithm. **Journal of the Royal Statistical Society**. Series B (methodological), v. 39, n.1, p. 1-38, 1977.

DOMINGOS, P.; HULTEN, G. Mining high-speed data streams. Em: **Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining**. ACM, 2000. p. 71-80.

FAROOQ, A.; ANWAR, S.; AWAIS, M.; ALNOWAMI, M. Artificial intelligence based smart diagnosis of Alzheimer's disease and mild cognitive impairment. Em: **Smart Cities Conference (ISC2), 2017 International**. IEEE, 2017. p. 1-4.

FONG, S.; ZHANG, Y.; FIAIDHI, J.; MOHAMMED, O.; MOHAMMED,S. Evaluation of stream mining classifiers for real-time clinical decision support system: a case study of blood glucose prediction in diabetes therapy. **BioMed Research International**, v. 2013, 2013.

FOLSTEIN, M. F.; FOLSTEIN, S. E.; MCHUGH, P. R. Mini-mental state: A practical method for grading the cognitive state of patients for the clinician. **J. Psychiatr. Res.**, v. 12, n. 3, p. 189-98, 1975.

GAO, X.W.; HUI, R. A deep learning based approach to classification of CT brain images. Em: **SAI Computing Conference (SAI), 2016**. IEEE, 2016. p. 28-31.

GARCÍA, J.M.M.; BÁEZ, P.G.; PINO, M.A.P.; VIADERO, C.F.; SUÁREZ-ARAUJO, C.P. A Counterpropagation network based system for screening of Mild Cognitive Impairment. Em: **Intelligent Systems and Informatics (SISY), 2012 IEEE 10th Jubilee International Symposium on**. IEEE, 2012. p. 67-72.

GUERRERO, J. M.; MARTÍNEZ-TOMÁS, R.; RINCÓN, M.; PERAITA, H. Diagnosis of Cognitive Impairment Compatible with Early Diagnosis of Alzheimer's Disease. **Methods of Information in Medicine**, v. 55, n. 01, p. 42-49, 2016.

GRABER, M.L.; FRANKLIN, N.; GORDON, R. Diagnostic error in internal medicine. **Archives of Internal Medicine**, v. 165, n. 13, p. 1493-1499, 2005.

HALL, P.M.; MARSHALL, A. D.; MARTIN, R.R. Incremental eigenanalysis for classification. In: **BMVC**. 1998. p. 286-295.

HAN, J., KAMBER, M., PEI, J. **Data Mining: Concepts and Techniques**. Elsevier, 3° edição, 2011.

HAYNES, R. Brian; WILCZYNSKI, Nancy L. Effects of computerized clinical decision support systems on practitioner performance and patient outcomes: methods of a decision-maker-researcher partnership systematic review. **Implementation Science**, v. 5, n. 12, p. 1-8, 2010.

HOSSEINI-ASL, E.; KEYNTO, R.; EL-BAZ, A. Alzheimer's disease diagnostics by adaptation of 3D convolutional network. **arXiv preprint arXiv:1607.00455**, 2016.

HU, C.; JU, R.; SHEN, Y.; ZHOU, P.; LI, Q. Clinical decision support for Alzheimer's disease based on deep learning and brain network. In: **Communications (ICC), 2016 IEEE International Conference on**. IEEE, 2016. p. 1-6.

HUANG, W.; CHEN, G. A novel functional MRI-based immersive tool for dementia disease severity prediction via spectral clustering and incremental learning. Em: 2015 **International Conference on Orange Technologies (ICOT)**. IEEE, 2015. p. 169-172.

ILLÁN, I; GÓRRIZ, J.; LÓPEZ, M.; RAMÍREZ, J.; SALAS-GONZALEZ, D.; SEGOVIA, F.; CHAVES, R.; PUNTONET, C.G. Computer aided diagnosis of Alzheimer's disease using component based SVM. **Applied Soft Computing**, v. 11, n. 2, p. 2376-2382, 2011.

ILLÁN, I.A.; GÓRRIZ, J.M.; RAMÍREZ, J.; LANG, E.W.; SALAS-GONZALEZ, D.; PUNTONET, C.G. Bilateral symmetry aspects in computer-aided Alzheimer's disease diagnosis by single-photon emission-computed tomography imaging. **Artificial Intelligence in Medicine**, v. 56, n. 3, p. 191-198, 2012.

JOHN, G.H.; LANGLEY, P. Estimating continuous distributions in Bayesian classifiers. Em: **Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence**. Morgan Kaufmann Publishers Inc., 1995. p. 338-345.

JORM, A. F. The Informant Questionnaire on cognitive decline in the elderly (IQCODE): a review. **International Psychogeriatrics**, v. 16, n. 03, p. 275-293, 2004.

KASSEM, N.; TEAM, E. **Designing enterprise applications: Java 2 platform**. Addison-Wesley Longman Publishing Co., Inc., 2000.

KATZ, S. Assessing self-maintenance: activities of daily living, mobility, and instrumental activities of daily living. **Journal of the American Geriatrics Society**, v. 31, n. 12, p. 721-727, 1983.

KEITH, M.; SCHINCARIOL, M. **Pro EJB 3: Java Persistence API**. Apress, 2006.

KOLLER, D.; FRIEDMAN, N.; BACH, F. **Probabilistic graphical models: principles and techniques**. MIT press, 2009.

KONONENKO, I. On biases in estimating multi-valued attributes. **International Joint Conference on Artificial Intelligence**, p. 1034–1040. Lawrence Erlbaum Associates, Montreal, Quebec, Canada, 1995.

KORB, K. B.; NICHOLSON, A. E. **Bayesian Artificial Intelligence**. New York: CRC Press, 2° edição, 2011.

KRASHENYI, I.; POPOV, A.; RAMIREZ, J.; GORRIZ, J.M. Application of fuzzy logic for Alzheimer's disease diagnosis. Em: **Signal Processing Symposium (SPSympo), 2015**. IEEE, 2015. p. 1-4.

LANGSETH, H; NIELSEN, T. D.; RUMÍ, R e SALMERÓN, A. Inference in Hybrid Bayesian Networks. **Reliability Engineering & System Safety**, v. 94, n. 10, p. 1499-1509, 2009.

LAURITZEN, Steffen L.; SPIEGELHALTER, David J. Local computations with probabilities on graphical structures and their application to expert systems. **Journal of the Royal Statistical Society. Series B (Methodological)**, p. 157-224, 1988.

LAWTON, M.P.; BRODY, E.M. Assessment of older people: self-maintaining and instrumental activities of daily living. **The Gerontologist**, v. 9, n. 3_Part_1, p. 179-186, 1969.

LEE, G.Y.; KIM, J.; KIM, J.H.; SEONG, J.K. Online learning for classification of Alzheimer disease based on cortical thickness and hippocampal shape analysis. **Healthcare Informatics Research**, v. 20, n. 1, p. 61-68, 2014.

LI, F.; TRAN, L.; THUNG, K.H.; JI,S.; SHEN,D.; LI,J. A robust deep model for improved classification of AD/MCI patients. **IEEE Journal of Biomedical and Health Informatics**, v. 19, n. 5, p. 1610-1616, 2015.

LI, Y.; CUI, D. Improvement and application of MVC design patterns [J]. **Computer Engineering**, v. 9, p. 35, 2005.

LIU, S.; LIU, S.; CAI, W.; PUJOL, S.; KIKINIS, R.; FENG, D. Early diagnosis of Alzheimer's disease with deep learning. In: **Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on**. IEEE, 2014. p. 1015-1018.

LIU, S.; LIU,S.; CAI,W., CHE, H.; PUJOL, S.; KIKINIS, R.;FENG,D.; FULHAN, M.J.; ADNI. Multimodal neuroimaging feature learning for multiclass diagnosis of Alzheimer's disease. **IEEE Transactions on Biomedical Engineering**, v. 62, n. 4, p. 1132-1140, 2015.

MATTILA, J.; KOIKKALAINEN, J.; VIRKKI, A.; GILS, M.V.;LÖTJÖNEM, J.; ADNI. Design and application of a generic clinical decision support system for multiscale data. **IEEE Transactions on Biomedical Engineering**, v. 59, n. 1, p. 234-240, 2012.

MIRHEIDARI, B.; BLACKBURN, D.; HARKNESS, K.; WALKER, T.; VENNERI, A.; REUBER, M.; CHRISTENSEN, H. Toward the Automation of Diagnostic Conversation Analysis in Patients with Memory Complaints. **Journal of Alzheimer's Disease**, v. 58, n. 2, p. 373-387, 2017.

MIYAMOTO, S.T.; JÚNIOR, I.L.; BERG, K.O.; RAMOS, L.R.; NATOUR, J. Brazilian version of the Berg balance scale. **Brazilian Journal of Medical and Biological Research**, v. 37, n. 9, p. 1411-1421, 2004.

MORABITO, F.C.; CAMPOLO, M.; IERACITANO, C.; EBADI, J.M.; BONANNO, L.; BRAMANTI, A.; DESALVO, S.; MAMMONE, N.; BRAMANTI, P. Deep convolutional neural networks for classification of mild cognitive impaired and Alzheimer's disease patients from scalp EEG recordings. Em: **Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016 IEEE 2nd International Forum on**. IEEE, 2016. p. 1-6.

MOREIRA, L. B.; NAMEN, A. A. System predictive for Alzheimer's disease in clinical trial. **Journal of Health Informatics**, v. 8, n. 3, 2016.

MORRIS, J.C. The Clinical Dementia Rating (CDR): current version and scoring rules. **Neurology**, v. 43, n. 11, p. 2412-2414, 1993.

NEWMAN-TOKER, D.E.; PRONOVOST, P.J. Diagnostic errors—the next frontier for patient safety. **Jama: the Journal of the American Medical Association**, v. 301, n. 10, p. 1060-1062, 2009.

NITRINI, R.; BOTTINO, C.M.; ALBALA, C.; CAPUNAY, N.S.C; KETZOIAN, C.; RODRIGUES, J.J.L.; MAESTRE, G.E.; RAMOS-CERQUEIRA, A.T.; CARAMELLI, P. Prevalence of dementia in Latin America: a collaborative study of population-based cohorts. **International Psychogeriatrics**, v. 21, n. 4, p. 622-630, 2009.

ORTIZ, A.; GÓRRIZ, J.M., RAMÍREZ, J.; MARTÍNEZ-MURCIA, F.J.; ADNI. LVQ-SVM based CAD tool applied to structural MRI for the diagnosis of the Alzheimer's disease. **Pattern Recognition Letters**, v. 34, n. 14, p. 1725-1733, 2013.

PANG, S.; OZAWA, S.; KASABOV, N. Incremental linear discriminant analysis for classification of data streams. **IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics** v. 35, n. 5, p. 905-914, 2005.

PASSOS, V. M. D. A.; GIATTI, L.; BARRETO, S. M.; FIGUEIREDO, R. C.; CARAMELLI, P.; BENSENOR, I.; DA FONSECA, M. J. M.; CCDE, N. V.; GOULART, A. C.; NUNES, M. A.; ALVES, M. G. D. M.; TRINDADE, A.A.M.D. Verbal fluency tests reliability in a Brazilian multicentric study, ELSA-Brasil. **Arquivos de Neuro-Psiquiatria**, v. 69, p. 814-816, 2011.

PAYAN, A.; MONTANA, G. Predicting Alzheimer's disease: a neuroimaging study with 3D convolutional neural networks. **arXiv preprint arXiv:1502.02506**, 2015.

PEARL, J. **Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference**. Morgan Kaufmann Publishers, Inc.,1988.

PEKKALA, T.; HALL, A.; LÖTJÖNEN,J.; MATTILA,J.; SOININEN, H.; NGANDU, T.; LAATIKAINEN, T.; KIVIPELTO, M.; SOLOMON, A. Development of a late-life dementia

prediction index with supervised machine learning in the population-based CAIDE study. **Journal of Alzheimer's Disease**, v. 55, n. 3, p. 1055-1067, 2017.

PEREIRA, T.; LEMOS, L.; CARDOSO, S.; SILVA, D.; RODRIGUES, A.; SANTANA, I.; MENDONÇA, A.; GUERREIRO, M.; MADEIRA, S.C. Predicting progression of mild cognitive impairment to dementia using neuropsychological data: a supervised learning approach using time windows. **BMC Medical Informatics and Decision Making**, v. 17, n. 1, p. 110, 2017.

PFEFFER, R. I.; KUROSAKI, T. T.; HARRAH, C. H.; CHANCE, J. M.; FILOS, S. Measurement of functional activities in older adults in the community. **Journal of Gerontology**, v. 37, n. 3, p. 323-329, 1982.

PLATT, J.C. 12 fast training of support vector machines using sequential minimal optimization. **Advances in Kernel Methods**, p. 185-208, 1999.

PRADHAN, M.; PROVAN, G.; MIDDLETON, B.; HENRION, M. Knowledge engineering for large belief networks. Em: **Proceedings of the Tenth Conference of Uncertainty in Artificial Intelligence.** Morgan Kaufmann Publishers Inc., 1994. p. 484–490.

QUINLAN, J.R. Induction of decision trees. **Machine Learning**, v. 1, n. 1, p. 81-106, 1986.

QUINLAN, J. **C4.5: programs for machine learning**. Morgan Kaufmann Publishers, Inc., 1993.

QUIONERO-CANDELA, J.; SUGIYAMA, M.; SCHWAIGHOFER, A.; LAWRENCE, N. D. **Dataset shift in machine learning**. The MIT Press, 2009.

REZAEI, M.; YANG, H.; MEINEL, C. Brain Abnormality Detection by Deep Convolutional Neural Network. **arXiv preprint arXiv:1708.05206**, 2017.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. Why should I trust you?: Explaining the predictions of any classifier. Em: **Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining**. ACM, 2016. p. 1135-1144.

ROBERTSON, G.; LEHMANN, E.D.; SANDHAM, W.; HAMILTON, D. Blood glucose prediction using artificial neural networks trained with the AIDA diabetes simulator: a proof-of-concept pilot study. **Journal of Electrical and Computer Engineering**, v. 2011, p. 2, 2011.

ROSADO, L.; VASCONCEDLOS, M.J.M.; CORREIA, F.; COSTA, N. A novel framework for supervised mobile assessment and risk triage of skin lesions. Em: **Proceedings of the 9th International Conference on Pervasive Computing Technologies for Healthcare**. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2015. p. 266-267.

SALAS-GONZALEZ, D.; GARRIZ, J.M.; RAMIREZ, J.; LOPEZ, M.;ALVAREZ, I.; SEGOVIA, F.; CHAVEZ, R., PUNTONET, C.G. Computer-aided diagnosis of Alzheimer's disease using support vector machines and classification trees. **Physics in Medicine and Biology**, v. 55, p. 2807-2817, 2010.

SARRAF, S.; TOFIGHI, G. Deep learning-based pipeline to recognize Alzheimer's disease using fMRI data. Em: **Future Technologies Conference (FTC)**. IEEE, 2016. p. 816-820.

SEGOVIA, F.; GÓRRIZ, J.; RAMÍREZ,J; SALAS-GONZALEZ, D.; ÁLVAREZ, I. Early diagnosis of Alzheimer's disease based on partial least squares and support vector machine. **Expert Systems with Applications**, v. 40, n. 2, p. 677-683, 2013.

SEGOVIA, F.; BASTIN, C.; SALMON, E.; GÓRRIZ, J.M.; RAMÍREZ, J.; PHILIPS, C. Combining PET images and neuropsychological test data for automatic diagnosis of Alzheimer's disease. **PLoS One**, v. 9, n. 2, p. 1-8, 2014.

SEIXAS F.L.; ZANDROZNY B.; LAKS, J.; CONCI, A.; MUCHALUAT-SAADE, D. C. A Bayesian network decision model for supporting the diagnosis of dementia, Alzheimer's disease and mild cognitive impairment. **Computers in Biology and Medicine**, v. 51, p. 140-58, 2014.

SERHANI, M.A.; BENHARREF, A.; NUJUM, A.R. Intelligent remote health monitoring using evident-based DSS for automated assistance. Em: **Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE**. IEEE, 2014. p. 2674-2677.

SHEIK J. I.; YESAVAGE, J. A. Geriatric Depression Scale (GDS): recent evidence and development of a shorter version**. Clinical Gerontologist: The Journal of Aging and Mental Health,** v. 5, n. 1-2, p. 165-173, 1986.

SHULMAN, K. I.; SHEDLETSKY, R.; SILVER, I.L. The challenge of time: clock-drawing and cognitive function in the elderly. **International Journal of Geriatric Psychiatry**, v. 1, n. 2, p. 135-140, 1986.

SINGH, L.; CHETTY, G. A comparative study of MRI data using various machine learning and pattern recognition algorithms to detect brain abnormalities. Em: **Proceedings of the Tenth Australasian Data Mining Conference-Volume 134**. Australian Computer Society, Inc., 2012. p. 157-165.

SONG, L.; HSU, W.; XU, J.; SCHAAR, M.V.D. Using contextual learning to improve diagnostic accuracy: Application in breast cancer screening. **IEEE Journal of Biomedical and Health Informatics**, v. 20, n. 3, p. 902-914, 2016.

SOUNTHARRAJAN, S.; THANGARAJ, P. Optimized feature selection technique for automatic classification of MRI images for Alzheimer's disease. **Journal of Medical Imaging and Health Informatics**, v. 6, n. 8, p. 2057-2062, 2016.

SOSA-ORTIZ, A.L.; ACOSTA-CASTILLO, I.; E PRINCE, M.J. Epidemiology of dementias and Alzheimer's disease. **Archives of Medical Research**, v. 43, n. 8, p. 600–608, 2012.

STROOP, J. R. Studies of interference in serial verbal reactions. **Journal of Experimental Psychology: General**, v. 121, n. 1, p. 15, 1992.

SUK, H.l.; SHEN, D. Deep learning-based feature representation for AD/MCI classification. Em: **International Conference on Medical Image Computing and Computer-Assisted Intervention**. Springer, Berlin, Heidelberg, 2013. p. 583-590.

SUN, Y.; LV, S.; TANG, Y. Construction and application of Bayesian network in early diagnosis of Alzheimer's disease. Em: **Complex Medical Engineering, 2007. CME 2007. IEEE/ICME International Conference on**. IEEE, 2007. p. 924-929.

TAN, Y.; SHENOY, P. P.; CHAN, M. W.; ROMBERG, P. M. On Construction of Hybrid Logistic Regression-Naïve Bayes Model for Classification. Em: **Conference on Probabilistic Graphical Models**, 2016. p. 523-534.

TOMBAUGH, T.N. Trail Making Test A and B: normative data stratified by age and education. **Archives of Clinical Neuropsychology**, v. 19, n. 2, p. 203-214, 2004.

THORNTON, C.; HUTTER, F.; HOOS, H.H.; LEYTON-BROWN, K. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms.

Em: **Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining**. ACM, 2013. p. 847-855.

TRAMBAIOLLI, L.R.; LORENA, A.C.; FRAGA, F.J.; KANDA, P.A.; ANGHINAH, R.; NITRINI, R. Improving Alzheimer's disease diagnosis with machine learning techniques. **Clinical EEG and Neuroscience**, v. 42, n. 3, p. 160-165, 2011.

TRIPOLITI, E.E.; FOTIADIS, D. I.; ARGYROPOULOU, M. A supervised method to assist the diagnosis and monitor progression of Alzheimer's disease using data from an fMRI experiment. **Artificial Intelligence in Medicine**, v. 53, n. 1, p. 35-45, 2011.

ÜSTÜN, B.; MELSSEN, W.J.; BUYDENS, L.M.C. 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.

VELERT, S. T. **Incremental Learning approaches to Biomedical decision problems**. Tese de Doutorado. Departamento de Física Aplicada, Universidad Politécnica de Valencia. 2012.

WEBB, G. I.; BOUGHTON, J. R., WANG, Z.: Not so Naïve Bayes: aggregating one-dependence estimators. **Machine Learning**, v. 58, n. 1, p. 5-24, 2005.

WEAKLEY, A.; WILLIAMS, J.A.; SCHMITTER-EDGECOMBE, M.; COOK, D.J. Neuropsychological test selection for cognitive impairment classification: a machine learning approach. **Journal of Clinical and Experimental Neuropsychology**, v. 37, n. 9, p. 899-916, 2015.

WORLD HEALTH ORGANIZATION (WHO). Mental health and older adults. Geneva: World Health Organization, 2013. Disponível em: <http://www.who.int/mediacentre/factsheets/fs381/en/>. Acesso em: 26 de junho de 2017.

YANG, H.; FONG, S. Incremental optimization mechanism for constructing a decision tree in data stream mining. **Mathematical Problems in Engineering**, v. 2013, p. 1-14, 2013.

YENER, M.; THEEDOM, A. **Professional Java EE Design Patterns**. John Wiley & Sons, 2014.

ZERBINI, F.M.G.; CALOMENI, M.R.; SILVA, V.F.; CABRAL, H.W.S; SCHIMIDT, S.L. Eficácia do teste Computadorizado de Atenção Visual (TCA) e do Mini Exame do Estado Mental (MEEM) na identificação de portadores de Transtorno Cognitivo Leve. **Pensar a Prática**, v. 12, n. 2, 2009.

ZHANG, Y.; FONG, S.; FIAIDHI, J.; MOHAMMED, S. Real-time clinical decision support system with data stream mining. **BioMed Research International**, v. 2012, 2012.

ZHOU, Q.; GORYAWALA, M.; CABRERIZO, M.; WANG, J.; BARKER, W.; LOEWENSTEIN, D. A.; DUARA, R.; ADJOUADI, M. An optimal decisional space for the classification of Alzheimer's disease and mild cognitive impairment. **IEEE Trans. Biomed. Engineering**, v. 61, n. 8, p. 2245-2253, 2014.