Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Uma Abordagem Computacional de Predição de Desempenho Acadêmico de Estudantes em Cursos *Online* de Programação

Fabrísia Ferreira de Araújo

CAMPINA GRANDE - PB
MAIO – 2019

Fabrísia Ferreira de Araújo

Uma Abordagem Computacional de Predição de Desempenho Acadêmico de Estudantes em Cursos *Online* de Programação

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande – Campus I como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Orientador:
Prof. Dr. Leandro Dias da Silva

Co-Orientador:
Prof. Dr. Evandro de Barros Costa

CAMPINA GRANDE - PB
MAIO – 2019

A663a Araújo, Fabrísia Ferreira de.

Uma abordagem computacional de predição de desempenho acadêmico de estudantes em cursos online de programação / Fabrísia Ferreira de Araújo. – Campina Grande, 2019.

146 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2019.

"Orientação: Prof. Dr. Leandro Dias da Silva, Prof. Dr. Evandro de Barros Costa".

Referências.

1. Inteligência Artificial. 2. Educação Online. 3. Mineração de Dados Educacionais. 4. Modelos Preditivos. I. Silva, Leandro Dias da. II. Costa, Evandro de Barros. III. Título.

CDU 004.8:37.018.43(043)

FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECÁRIA SEVERINA SUELI DA SILVA OLIVEIRA CRB-15/225

"UMA ABORDAGEM COMPUTACIONAL DE PREDIÇÃO DE DESEMPENHO ACADÊMICO DE ESTUDANTES EM CURSOS ONLINE DE PROGRAMAÇÃO"

FABRÍSIA FERREIRA DE ARAÚJO

TESE APROVADA EM 23/05/2019

LEANDRO DIAS DA SILVA, Dr., UFAL
Orientador(a)

EVANDRO DE BARROS COSTA, Dr., UFAL
Orientador(a)

JOSEANA MACÊDO FECHINE RÉGIS DE ARAÚJO, Dra., UFCG
Examinador(a)

DALTON DARIO SEREY GUERRERO, Dr., UFCG
Examinador(a)

EDILSON FERNEDA, Dr., UCB
Examinador(a)

PATRICK HENRIQUE DA SILVA BRITO, Dr., UFAL
Examinador(a)

MARÍA DEL ROSARIO GIRARDI GUTIERREZ, Dra., UFMA
Examinador(a)

CAMPINA GRANDE - PB

Agradecimentos

Inicialmente gostaria de agradecer a Deus, por guiar e iluminar minha vida em momentos de dúvidas,de dor, de sobrecargas, de falta de motivação; por sua providência em colocar pessoas que me ajudaram neste percurso. Agradeço aos meus pais, pelo apoio na vida e particularmente nesta volta como estudante a Campina Grande; aos meus filhos, que sem querer me impulsionam e motivam a estar sempre em movimento, buscando ser espelho pra eles; ao meu esposo, pelo suporte científico, atenção, paciência e crença em minhas capacidades. Ao Ifal, pelo suporte financeiro.

Aos colegas que de alguma forma estiveram comigo nessa caminhada compartilhando experiências e estudos, aos funcionários da COPIN/UFCG, especialmente a profa. Dra. Livia Campos, e a todas as pessoas que de alguma forma contribuíram para esta pesquisa.

Aos orientadores Leandro e Evandro que me deram a oportunidade de desenvolver esta pesquisa, prestando caras e valiosas contribuições para que ela se materializasse nesta tese.

.

Resumo

Há um alto índice de insucesso em algumas disciplinas iniciais nos cursos de graduação em computação, tanto na modalidade presencial quanto a distância, notadamente em disciplinas de Programação e Matemática. Recentemente, tem se percebido como tendência que a execução de tais disciplinas na modalidade presencial seja associada a uma complementação *online*, tendo deste modo tal como ocorre na educação a distância, um ambiente virtual de aprendizagem vinculado, o qual viabiliza interações entre estudantes e professores, potencialmente produzindo uma grande quantidade de dados. Neste contexto, surge um problema de pesquisa amplo e importante que é o de como aplicar mineração de dados, via modelagem preditiva, para gerar informação de alta qualidade: correta, oportuna e útil, permitindo, por exemplo, subsidiar efetivamente o processo de decisões pedagógicas a serem tomadas por professores, possibilitando contribuir na redução do mencionado índice de insucesso. Neste sentido, o objetivo geral da presente pesquisa é propor uma abordagem preditiva para identificar, o mais cedo possível, estudantes com risco de insucesso acadêmico, focalizando-se disciplinas de programação introdutória, levando-se em consideração a confiabilidade e compreensibilidade dos modelos produzidos. Nesta abordagem, privilegiou-se modelos preditivos do tipo caixa branca, tendo em vista o potencial de tais modelos no atendimento ao requisito de compreensibilidade. Para avaliar a abordagem proposta, realizou-se vários estudos empíricos, utilizando-se dados de estudantes de uma instituição pública de ensino superior, associados a disciplinas de programação, além de dados socioeconômicos e demográficos. Deste modo, os resultados obtidos mostraram a viabilidade e efetividade da abordagem proposta, tanto no que diz respeito à qualidade da predição, quanto na indicação da influência dos atributos selecionados. Portanto, concluiu-se que o modelo é capaz de realizar predição antecipada com acurácia satisfatória dentro dos padrões comparativos da literatura e com boa interpretabilidade, sendo assim, útil para auxiliar em tomadas de decisões pedagógicas por parte dos professores.

Palavras-chave: Mineração de Dados Educacionais, Modelos Preditivos, Educação Online.

Abstract

There is a high failure rate in some initial disciplines in computing undergraduate courses, both in face-to-face and distance modalities, especially in programming and mathematics disciplines. Recently, it has been perceived as a tendency that the execution of such disciplines in the face-to-face modality has been associated to an online complementation, thus having, as it happens in distance education, a virtual learning environment linked, which enables interactions between students and teachers, potentially producing a large amount of data. In this context, a broad and important research problem arises that is how to apply data mining, via predictive modeling, to generate high quality information: correct, timely and useful, allowing, for example, effectively subsidize the process of pedagogical decisions to be taken by teachers, aiming to contribute to the reduction of the mentioned failure rate. In this sense, the overall objective of this research is to design and develop a predictive approach to identify, as soon as possible, students who may be at risk of failing or dropping introductory programming courses, taking into consideration to generate reliable and comprehensible prediction models. In this approach, we gave priority to white-box predictive models, appreciating that this type of model is potentially more adequate to offer comprehensibility in terms of information in the model. To evaluate the proposed approach, we conducted several empirical studies using academic data of students from a public university, as well as socioeconomic and demographic data. Thus, the obtained results showed the feasibility and effectiveness of the proposed approach, both with respect to the quality of the prediction, as well as in the indication of the influence of the selected attributes. Therefore, it was concluded that the model is interpretable and able to perform prediction with satisfactory accuracy within the comparative standards of the literature and, thus, it is useful to aid in pedagogical decision making by the teachers.

Keywords: Educational Data Mining, Prediction Model, Online Education.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

AVA – Ambientes Virtuais de Aprendizagem
EaD – Educação a Distância
IA – Inteligência Artificial
IFAL – Instituto Federal de Educação, Ciência e Tecnologia de Alagoas
UFAL- Universidade Federal de Alagoas
LDB - Lei de Diretrizes e Bases da Educação Nacional
MEC – Ministério da Educação
MOOC – *Massive Open Online Course*
RS – Revisão Sistemática
UAB - Universidade Aberta do Brasil
ENEM- Exame Nacional do Ensino Médio
TIPS - Tecnologias Inteligentes, Personalizadas e Sociais. (Grupo de Pesquisa da UFAL, cadastrado no CNPq).

Sumário

1 INTRODUÇÃO

A presente pesquisa aborda a questão de desempenho acadêmico de estudantes, focalizando principalmente aspectos de retenção de tais estudantes em determinadas disciplinas iniciais de um curso. Situa-se tecnicamente em uma interseção dos domínios de Inteligência Artificial em Educação e Descoberta de Conhecimento em Bases de Dados (FAYYAD et al., 1996), utilizando-se de técnicas de mineração de dados, considerando-se principalmente dados acadêmicos obtidos em plataformas de aprendizagem *online*, além de alguns dados socioeconômicos e demográficos. Tais plataformas podem ser caracterizadas principalmente por prover facilidades para interações envolvendo agentes humanos e agentes de software, imersos em um ambiente com conteúdos digitais. Neste contexto, esta pesquisa se insere em um projeto maior no qual se pretende desenvolver uma abordagem computacional de suporte à decisão baseada em dados, destinando-se ao provimento de mecanismos inteligentes e adaptativos de apoio a estudantes e professores, atuando em ambientes virtuais de aprendizagem no domínio de programação introdutória. Na pesquisa ora proposta, entretanto, investigou-se a utilização de técnicas de mineração de dados educacionais a fim de construir, notadamente, modelos preditivos de desempenho acadêmico de estudantes, possibilitando-se auxiliar professores com informações valiosas em suas tomadas de decisões pedagógicas.

Neste capítulo, apresenta-se o tema no qual se insere a presente pesquisa. Assim, discute-se uma motivação e contextualização do assunto, além da problemática central e objetivo proposto, aspectos metodológicos, incluindo ainda uma primeira declaração de delimitação da pesquisa e principais contribuições esperadas da tese, finalizando-se com a descrição da organização desta tese.

1.1 Motivação e Contextualização da Pesquisa

Nos últimos anos, o domínio de Educação, em todos os níveis e modalidades,

vem passando por várias transformações, notadamente pela forte influência que vem tendo do uso das novas tecnologias da informação e comunicação, particularmente pela adoção de plataformas de aprendizagem online e disponibilização de recursos digitais cada vez mais sofisticados. Assim, em um sentido mais amplo, pode-se denominar de educação online, toda uma variedade de oferta educacional fazendo uso de algum ambiente virtual de aprendizagem, permitindo aos participantes, por exemplo, flexibilidades temporais e espaciais, diferentes tipos de interação, acesso e compartilhamento de recursos com os participantes. Neste contexto online, há ambientes virtuais de aprendizagem que disponibilizam ainda, aos seus usuários, várias outras facilidades, tais como o oferecimento de serviços educacionais que se prestam a potencializar formas de personalização da aprendizagem, visando oferecer uma educação com mais qualidade.

Em particular, a educação superior, tanto na modalidade a distância quanto na presencial, vem seguindo a perspectiva de educação online mencionada no parágrafo anterior, adotando-se plataformas de aprendizagem online e, assim, tendo as condições adequadas para se produzir um grande volume de dados, oriundo de interações dos usuários, tendo, deste modo, potenciais informações sobre o comportamento dos estudantes (SHAHIRI et al., 2015; CONIJN et al., 2017; ZAFFAR et al., 2018). No entanto, por exemplo, há ainda estatísticas alarmantes com respeito aos índices de reprovação em disciplinas no ciclo básico de cursos de graduação, especificamente, nas áreas de ciências exatas, incluindo-se engenharias e computação. Particularmente, tal cenário se verifica marcantemente nas disciplinas iniciais de Programação e Patemática (WATSON & LI, 2014; BENNEDSEN & CASPERSEN, 2007; MCGETTRICK, A. et al, 2005; COSTA et al., 2017), nas quais se constatam um alto índice de insucesso por parte dos estudantes ao enfrentar estas disciplinas nos cursos de graduação em computação, tanto na modalidade presencial quanto a distância, o que em determinados casos culmina com evasão de parte destes estudantes (TAN et al., 2009; IEPSEN et al., 2013; CAMBRUZZI et al., 2015; MARQUEZ-VERA et al., 2016).

O cenário de insucesso anteriormente descrito, mostra-se propício à utilização de técnicas preditivas na área de mineração de dados educacionais,

prestando-se para fornecer indícios importantes de como os estudantes interagem e aprendem (SLATER et al.,2016; KOEDINGER et al., 2013), potencialmente percebendo as individualidades de cada estudante, permitindo intervenções pedagógicas adaptativas. Portanto, a presente pesquisa se situa na linha de mineração de dados educacionais, focalizando principalmente a construção de modelos preditivos de desempenho acadêmico de estudantes, pretendendo-se gerar informações relevantes para, em última instância, auxiliar professores ou agentes de software tutores, em suas tomadas de decisões pedagógicas.

1.2 Problemática

No contexto anunciado na Seção 1.1, mencionou-se o desafio associado a como reduzir os altos índices de retenção, bem como perguntas que surgem em cenários de educação online que potencialmente produz um grande volume de dados gerado nas interações dos estudantes, ao tentar explorar tais dados. Nisso, localiza-se um amplo problema de pesquisa que é o de como conceber mecanismos automáticos capazes de analisar dados educacionais, principalmente, mas não unicamente, os produzidos nas interações dos estudantes com o ambiente online, visando a geração de informação de qualidade, cumprindo os requisitos de ser relevante, confiável e oportuna, sendo produzida a um custo adequado, subsidiando o processo de decisões pedagógicas de professores e de mecanismos automáticos, tendo em conta, em última instância, a redução dos índices de insucesso acadêmico de estudantes. A ideia é a de que tais mecanismos possam informar o mais cedo possível com alertas sobre estudantes com tendências ao insucesso em termos de desempenho acadêmico, a fim de que o professor possa se antecipar aos problemas, oferecendo auxílio apropriado a tais estudantes.

O problema descrito no parágrafo anterior é muito amplo e, deste modo, vem sendo desdobrado em vários subproblemas concretos, incluindo-se o da

predição antecipada[1] e seus desdobramentos em busca de eficiência e explicabilidade do modelo. Este problema tem sido perseguido em diferentes trabalhos representativos do atual estado da arte (HU et al., 2014; MARQUEZ-VERA et al., 2016; HUNG et al., 2016; MARBOUTI et al, 2016; COSTA et al., 2017; CHUI et al, 2018).

Na perspectiva mencionada anteriormente, a presente investigação científica buscou responder aos seguintes problemas norteadores desta pesquisa: Como desenvolver um modelo preditivo, com desempenho satisfatório, permitindo identificar sucesso acadêmico de estudantes, no tempo mais cedo possível e em diferentes períodos de tempo posteriores, a partir de dados desses estudantes, considerando-se dados disponíveis imediatamente antes de iniciar o curso e, principalmente, os dados acadêmicos obtidos ao longo do curso, bem como dados disponíveis imediatamente antes de iniciar o curso? Como assegurar que um tal modelo preditivo possa ser adequado para apoiar o professor na tomada de decisão pedagógica, levando-se em consideração uma solução de compromisso entre uma acurácia satisfatória e a compreensibilidade do modelo?

A partir destes dois problemas mencionados, definiu-se o objetivo geral desta tese, tal como segue.

1.3 Objetivos

O objetivo geral desta pesquisa foi contribuir no estudo e desenvolvimento de modelos preditivos de desempenho acadêmico de estudantes de programação, propondo-se uma abordagem para identificar estudantes em situação de risco de insucesso, tão cedo quanto possível, bem como em momentos posteriores, endereçando, deste modo, os problemas de pesquisa norteadores anteriormente anunciados. Tal abordagem, portanto, visou desenvolver mecanismos preditivos capazes de predizer em diferentes períodos de tempo de um curso, o desempenho acadêmico de estudantes, em cursos de

[1] Embora o termo predição esteja intrinsecamente ligado à antecipação, usaremos aqui o termo predição antecipada como a tradução livre do termo inglês *early prediction*, significando predição mais cedo possível.

programação introdutória, considerando-se um momento de boa antecedência, tendo significativa precisão e compreensão adequada do modelo.

A fim de que o objetivo geral descrito fosse alcançado, os seguintes objetivos específicos foram delineados:

Obj1: Realizar um estudo empírico sobre o comportamento dos modelos preditivos selecionados, verificando seus desempenhos na tarefa de identificação, tão cedo quanto possível, de estudantes em risco de insucesso;

Obj2: Comparar os modelos estudados, considerando-se métricas de confiabilidade na predição;

Obj3: Identificar os principais atributos de um dado conjunto de dados, considerando-se suas relevâncias e níveis de influência para predição automática de desempenho acadêmico dos estudantes;

Obj4: Estudar alternativas para prover soluções baseadas em modelos de predição antecipada e temporal, buscando melhorar aspectos de efetividade em tais modelos;

Obj5: Desenvolver uma abordagem de predição em um tempo cedo, obtendo uma solução confiável e compreensível, tendo acurácia satisfatória, em momentos distintos, a saber: **Antes** de iniciar o curso e no **decorrer** do curso, no qual se persegue a predição antecipada

Obj6: Avaliar a abordagem preditiva proposta.

Ressalta-se que esses objetivos específicos ainda estão amplos, mas serão oportunamente desdobrados em mais especificidades nos capítulos 4 e 5 desta tese, ocasião na qual estarão associados a questões de pesquisa.

1.4 Aspectos Metodológicos

Esta pesquisa, de um modo geral, surgiu a partir da identificação de um problema envolvendo retenção e evasão de estudantes, no âmbito da educação online tanto na modalidade a distância quanto na presencial. Para

tanto, verificou-se a relevância das tarefas envolvidas, associadas ao problema, contextualizando-o mediante revisão da literatura, entrevistas com professores e análise de dados secundários. Assim, qualificou-se e formulou-se os problemas norteadores, bem como a maneira de abordá-los, estabelecendo-se questões de pesquisa, em seguida definindo-se os objetivos da tese.

Para realizar os objetivos propostos, seguiu-se nesta tese um percurso metodológico, em parte de natureza exploratória, que incluiu os seguintes grandes passos:

Passo 1: Estudo da literatura sobre retenção em cursos de programação e disciplinas similares, considerando-se o investimento em predição automática de desempenho acadêmico. Após a etapa de definição do tema e problema, um exame das bases de dados envolvidas, um estudo mais profundo dos conceitos essenciais relacionados a mineração de dados foi realizado. Neste passo, também foram analisados de forma mais minuciosa os trabalhos relacionados.

Passo 2: Especificação de uma solução ao problema abordado. Neste passo, foi especificada uma solução ao problema identificado no Passo 1, considerando-se uma análise exploratória, envolvendo um estudo comparativo minucioso das técnicas de mineração de dados para classificação, definindo-se um modelo preditivo para desempenho acadêmico.

Passo 3: Avaliação experimental da solução proposta. A última fase consistiu em avaliar o modelo desenvolvido no Passo 2. Isso foi feito através da definição e realização de quatro estudos de caso no domínio de programação que serviram para avaliar comparativamente a efetividade e o desempenho de cada classificador.

Passo 4: Verificar possibilidade de generalização da solução proposta a outros contextos ou situações.

1.5 Delimitação e Contribuições da Pesquisa

Esta pesquisa pretende contribuir para melhoria na formação dos estudantes

de graduação, mas possui um escopo que inicialmente se restringe à aplicação direta a domínios formais, notadamente programação, pois em tais domínios pode-se ter maior abrangência na elaboração de mecanismos automáticos de assistência ao estudante. No mais, tais domínios possuem uma grande relevância para o presente trabalho devido ao fato de serem associados a um alto índice de insucesso acadêmico, tal como já mencionado. Especificamente, a pesquisa está focada diretamente na disciplina de programação introdutória em cursos de graduação nas modalidades presencial e a distância, focalizado atividades no ambiente online. Há outras restrições mais específicas que são oportunamente comentadas ao longo do texto.

Com respeito às contribuições, ressalta-se o modelo preditivo desenvolvido para identificação, tão cedo quanto possível, em diferentes períodos de tempo, de estudantes com propensão ao insucesso, possibilitando aos professores dispor de diagnósticos e realizar intervenções pedagógicas em tempos apropriados. Além disso, espera-se também contribuir para os estudantes e para os gestores, pois a informação produzida interessa a ambos. Portanto, pode contribuir para educação online.

1.6 Organização do Documento

Este documento está subdividido em seis capítulos, incluindo o presente capítulo introdutório e os demais que estão estruturados, conforme descrição a seguir.

No Capítulo 2, consta-se da fundamentação teórica da pesquisa em pauta, apresentando um esclarecimento dos principais conceitos e técnicas relacionados ao tema pesquisa em pauta, focalizando educação online e os ambientes virtuais de aprendizagem, assim como algumas técnicas de mineração de dados.

No Capítulo 3, trata-se da caracterização do estado da arte, enfatizando um elenco de trabalhos relacionados e posicionando melhor o presente trabalho.

No Capítulo 4, apresenta-se uma descrição detalhada de uma primeira investida para compor a solução proposta, destacando-se um estudo comparativo e de viabilidade do uso de técnicas de mineração de dados para predizer desempenho acadêmico de estudantes, envolvendo etapas de preparação dos dados e seleção de atributos, bem como a construção e avaliação de modelos preditivos.

No Capítulo 5, apresenta-se, com base em lições aprendidas no estudo no Capítulo 4, a abordagem desenvolvida, estando discutidos a metodologia e os resultados de avaliação dos modelos, verificando a acurácia das predições, além de uma análise dos resultados obtidos, incluindo aspectos de interpretabilidade dos modelos.

No Capítulo 6, apresentam-se as considerações finais acerca dos resultados obtidos na pesquisa, suas principais contribuições, limitações e os trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, apresenta-se uma descrição dos aspectos conceituais necessários à compreensão do tema da pesquisa em pauta. Assim, inicia-se com uma discussão sobre educação online e ambientes computacionais de suporte às atividades educacionais, ou seja, plataformas de aprendizagem online. Em seguida, faz-se uma discussão sobre descoberta de conhecimento em bases de dados e mineração de dados, fornecendo-se aspectos técnicos envolvidos, prestando-se a apoiar o entendimento da abordagem proposta.

2.1 Educação Online

Em um sentido mais amplo, o termo educação *online* pode ser entendido como uma forma de educação que é realizada, em algum nível, usando a Internet, deste modo podendo ser dividida em diferentes categorias. Tal divisão é feita com base no grau de aprendizagem online que é incorporada no curso, assim, variando de cursos presencias com algum uso de recurso *online* integrado, passando pela aprendizagem híbrida (blended), até cursos que são executados exclusivamente de forma *online*.

Ultimamente a realização de educação online tem sido incrementada devido à evolução dos meios de coleta e armazenamento de dados digitais, além do desenvolvimento de mecanismos ágeis para análise de grandes volumes de dados. No entanto, todo esse avanço nem sempre tem contribuído para redução de altos índices de reprovação em disciplinas ou mesmo de evasão nos cursos. Particularmente, em disciplinas iniciais de programação, em muitos casos, verifica-se um alto índice de retenção.

Há uma preocupação crescente com o processo de ensino e aprendizagem de programação, ao mesmo tempo em que a habilidade de programação tem sido cada vez mais valorizada em educação, chegando inclusive à educação no ensino fundamental e médio. A atividade de programação de computadores pode ser caracterizada como sendo resolução de problema, dado que um programa é a expressão de uma solução para um

determinado problema. Neste sentido, alguns estudos sobre educação em computação revelam que a aprendizagem de programação é uma tarefa difícil, requerendo dos estudantes boa habilidade para resolução de problemas e mais motivação, tendo mostrado altos índices de reprovação.

Recentemente, a execução de disciplinas de programação na modalidade presencial tem sido associada, ainda não muito amplamente, a uma complementação online, tendo, tal como na educação a distância, algum tipo de ambiente virtual de aprendizagem vinculado, dando suporte apropriado a estudantes e professores. Tal complementação potencializa mais opções para aprendizagem dos estudantes, ao mesmo tempo que possibilita a produção de uma grande variedade de dados educacionais gerados a partir das interações desses estudantes com recursos humanos ou digitais nos ambientes online de aprendizagem. Neste sentido, caracteriza-se a educação online, na qual configuram-se ambientes online de aprendizagem dotados de várias facilidades e ferramentas, prestando-se tanto à educação presencial quanto à distância, que possibilitam interações entre estudantes, professores e conteúdos digitais (AIRES, 2016). Deste modo, apresenta-se a Figura 2.1, como uma primeira abstração pictórica para ajudar a situar alguns dos principais problemas de modelagem de interação abordados neste trabalho, focalizando-se e analisando-se dados produzidos.

Figura 2.1 - Esquema geral das principais interações no SEIA

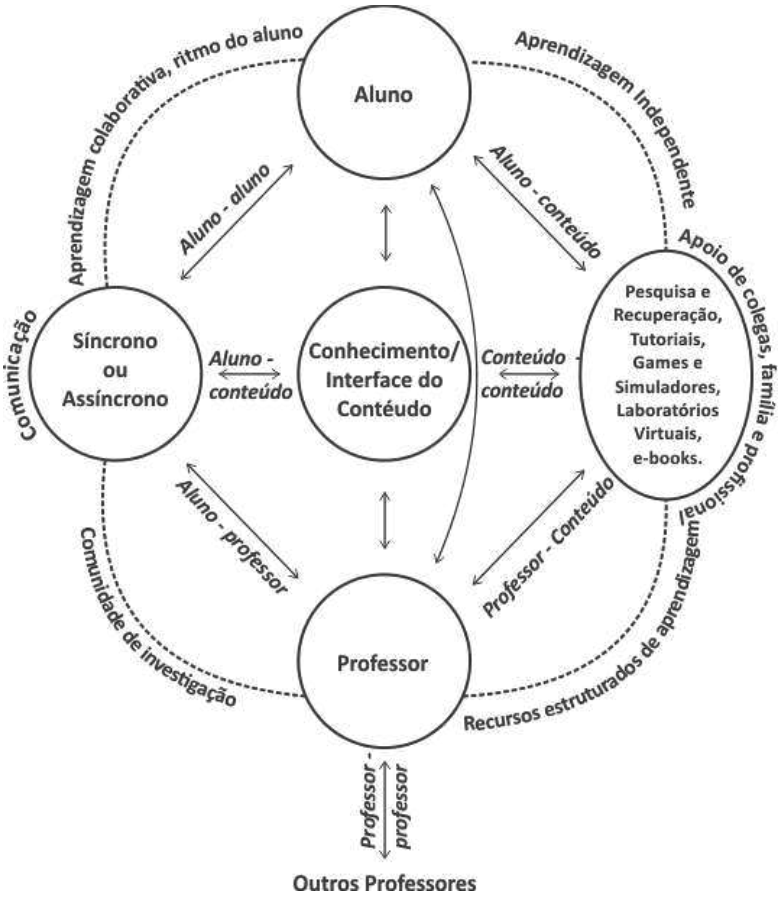
Fonte: Autora.

Na Figura 2.1, ilustra-se, portanto, um cenário típico no qual estudantes interagem com recursos de um ambiente virtual de aprendizagem dotado de agentes de software, professor e conteúdos digitais. Assim, pode-se proporcionar diversas situações de aprendizagem, possibilitando uma boa qualidade e quantidade interações entre os atores envolvidos e os conteúdos digitais disponíveis. Particularmente, estas interações produzem dados, os quais são armazenados e analisados por mecanismos computacionais, possibilitando-se deste modo gerar informações para os professores e para os agentes tutores. Neste ambiente, oferece-se aos estudantes suporte à resolução de problemas, podendo estender para outras atividades avaliativas. Assume-se aqui um esquema de aprendizagem centrada no estudante, permitindo mobilizar as seguintes interações: estudante e agente de software, estudante e recursos pedagógicos, estudante e agente humano (estudante, professor), agente de software e agente humano (estudante, professor), agente de software e recursos pedagógicos, agente humano e recursos pedagógicos. Este cenário sugere uma variedade de desafios envolvidos na modelagem das principais interações destacadas, representando aqui o contexto no qual se insere a presente pesquisa de doutorado, buscando-se uma abordagem para

solucionar alguns desses tais problemas.

O esquema da Figura 2.1 está em consonância com outros modelos conceituais, particularmente com o modelo pedagógico de aprendizagem online, proposto por ANDERSON, (2008), ressaltando-se o papel da interação no processo de aprendizagem mediada por computador, especificamente, na modalidade de educação a distância via Web. Segundo ANDERSON,(2004), a aprendizagem on-line é um fragmento de toda a educação a distância e contém características bastante peculiares que a tornam um modelo abrangente, com componentes bem definidos e as respectivas interações estabelecidas. Assim, neste modelo de aprendizagem on-line, apresentado na Figura 2.1, assim como no modelo da Figura 2.2, ilustram-se essencialmente os dois principais atores humanos: alunos e professores e suas interações entre si e com o conteúdo.

Figura 2.2 - Modelo de Aprendizagem Online[i]



Fonte: (ANDERSON, 2008).

Com base nestes dois modelos conceituais apresentados, pode-se

desenvolver implementações relevantes de ambientes educacionais online, sendo particularmente importantes para geração de dados e, assim, permitir a realização de análises de dados as mais sofisticadas. No entanto, neste contexto, ressalta-se que há carências importantes em muitas dessas iniciativas em desenvolvimento de ambientes, quanto à falta de uso de ferramentas inteligentes na assistência adaptativa aos estudantes e, notadamente, no suporte efetivo aos professores. Ainda há muito a se investir em tais ambientes, dotando-os de ferramentas mais inteligentes em favor dos estudantes e dos professores. Assim, vislumbra-se, particularmente, o desenvolvimento de ferramentas para mineração de dados educacionais, com o objetivo de incorporar técnicas de inteligência artificial ao projeto de um ambiente virtual de aprendizagem, ou alternativamente seguindo uma tendência de evolução dos sistemas tutores inteligentes que vem se desenhando na literatura de Inteligência Artificial em Educação (WOOLF et al., 2013), concretizando plenamente o que se tem denominado de ambientes virtuais de aprendizagem inteligentes.

2.2 Ambientes de Aprendizagem Online

A seguir estão discutidas algumas categorias de ambientes de aprendizagem online, selecionados pela pertinência ao que se utilizou no desenvolvimento da presente pesquisa, ou mesmo pela pertinência em já considerar o uso de técnicas de inteligência artificial. Ressalta-se que os ambientes descritos podem ser vistos como instância conceitual e realização tecnológica do modelo conceitual anteriormente exibido na Figura 2.2.

2.2.1 Ambientes Virtuais de Aprendizagem

Discute-se aqui, numa visão clássica, os Ambientes Virtuais de Aprendizagem (AVAs), os quais possuem também a denominação AVEA quando se pretende enfatizar a atividade de ensino, ficando Ambientes Virtuais de Ensino e Aprendizagem, ou ainda Sistemas de gerenciamento de aprendizagem. Na literatura inglesa, costuma-se utilizar as siglas LMS, para Learning Management System, ou ainda CMS, para Content Management System, ou

mesmo Virtual Learning Environment, ou ainda e-Learning platform. Esse tipo de sistema tem grande importância no impulso à educação online. Mas, além da menção a alguns exemplos, interessa-nos aqui primeiramente discutir como caracterizar um tal ambiente e enfatizar sua composição em termos de serviços disponibilizados para os principais atores usuários desses sistemas, a exemplo particularmente dos professores, estudantes, tutores, coordenadores de curso. Essencialmente tais ambientes focam no gerenciamento de conteúdos de aprendizagem e no suporte administrativo, provendo uma coleção de funcionalidades para serem usadas por esses atores.

Existem várias plataformas de Ambientes Virtuais de Aprendizagem (AVAs) disponíveis para uso gratuito, a exemplo do Moodle - Modular Object Oriented Distance Learning (https://moodle.org/) e Claroline (www.claroline.net/), e outros tantos utilizados de maneira privada, a exemplo do Pearson Learning Studio (http://developer.pearson.com/products/learningstudio), do WebCT (http://www.webct.com/) e Blackboard (http://www.blackboard.com/). No entanto, todos eles essencialmente partilham várias características e funcionalidades, entre as quais destacam-se conceitos, tais como:

Ator: Considera-se a existência e distinção dos atores que interagem com o ambiente, como, por exemplo, professor, estudante e tutor, permitindo customização de interface e serviços para cada um deles;

Serviço: Há uma variedade de serviços que podem ser associados a um determinado curso, destacando-se serviços administrativos, serviços didáticos (por exemplo, provendo gerenciamento, distribuição e compartilhamento de conteúdos de aprendizagem), serviços de avaliação, serviços de comunicação, serviços individuais, serviços de grupo;

Grupo: Este conceito trata da possibilidade de formação e organização de grupos entre os usuários do ambiente. Nele, pode-se identificar aspectos, tais como: definição do grupo: simples (não possui subgrupos) ou hierárquico (possui subgrupos); controle de visibilidade de grupos; e interação intergrupos: possibilidade de interação, cooperação e colaboração entre grupos.

Os AVAs dão suporte a atividades de ensino e aprendizagem e são

normalmente usados para dar suporte a um modelo de educação baseado em sala de aula. Entre seus objetivos, incluem-se o provimento de flexibilidade, acessibilidade e conveniência a seus usuários. Na presente pesquisa, fez-se uso do AVA Moodle em um dos estudos realizados

O conceito de AVA é algo ainda recente, mas já há estudos que ressaltam suas limitações mais importantes, destacando-se a falta de ferramentas inteligentes para monitoramento do ambiente, observando comportamento e necessidades dos estudantes. Um dos problemas apontados na literatura sobre os AVAs tradicionais tem sido ligado à tendência ao isolamento e a pouca participação de alguns estudantes no ambiente, sem que isso dispare algum mecanismo de feedback relevante por parte do sistema. Com o advento dos Cursos *Online* Massivos e Abertos (do inglês: Massive Open Online Courses – MOOCs) (SPOELSTRA et al., 2015), esse tipo de limitação passa a ser ainda mais critico, pois se passa a lidar com milhares de estudantes numa plataforma *online*, o que inclusive vem levando a investimentos de pesquisa em ferramentas da Inteligência Artificial (RUSSELL & NORVIG, 2009), a exemplo de técnicas de representação de conhecimento e raciocínio, aprendizagem automática, para serem integradas a tais ambientes. Como exemplo de plataformas para MOOCs, tem-se edX, UDacity, Coursera.

2.2.2 Juiz *Online*

Uma categoria de ambiente computacional que tem sido frequentemente usada em muitos cursos no suporte à aprendizagem de programação (RAADT, 2007) é o que se denomina de Juiz *online*. Trata-se de um recurso importante que oferece um mecanismo de avaliação automatizada das soluções submetidas pelos estudantes. Basicamente um juiz online oferece as seguintes funcionalidades: compilação do código submetido; execução do programa gerado; especificação de dados de entrada para a execução; e por fim a comparação da saída da execução com uma saída padrão esperada.

Tanto os dados de entrada quanto os dados de saída, são dados padrão da questão e são utilizados considerando que, se o algoritmo está correto, ele

gerará os dados de saída a partir da entrada definida. Esses dados não são de conhecimento dos usuários que submetem o algoritmo.

Há vários juízes *online* disponíveis para uso (PAES et al., 2013), a exemplo de URI Online Judge (urionlinejudge.com.br), UVA (uva.onlinejudge.org), The Huxley (Thehuxley.com), inclusive alguns já integrados a AVAs. Uma característica comum a todos é a disponibilidade de um suporte de avaliação automatizada da solução do problema expressa no código da linguagem em uso.

Por fim, convém particularmente destacar a importância desse tipo de ambiente no domínio de programação, contando especificamente com mecanismos de avaliação automatizada sobre as soluções via código. Isso é um ponto de partida interessante, mas ainda precisando evoluir na qualidade do que se avalia, assim como no que vem em decorrência disso. Particularmente o juiz *online* The Huxley foi utilizado na presente pesquisa, sendo constituído por uma ferramenta web que permite aos alunos submeter código-fonte em diversas linguagens de programação como resposta a exercícios de uma base de centenas de problemas de programação. Para cada submissão, o aluno recebe feedback da correção automática pelo sistema através de análise sintática do código e dos testes de aceitação. Foi pensado para auxiliar o aluno e professor dentro e fora de sala de aula (THEHUXLEY, 2017; PAES et al., 2013).

2.2.3 Sistema Tutor Inteligente

Um Sistema Tutor Inteligente (STI) representa uma categoria de software educativo que faz uso de técnicas de Inteligência Artificial para representação de conhecimento e raciocínio, visando assistir estudantes em suas atividades de aprendizagem, provendo-lhes suporte pedagógico adaptativo em um domínio de conhecimento particular (SLEEMAN AND BROWN, 1982; WOOLF, 2010). A pesquisa em STI iniciou-se na década de 70 (CARBONELI, 1970; SELF, 1974). Um STI, no sentido mais clássico, pretende reproduzir o comportamento atribuído a um professor humano atuando em um domínio de

conhecimento particular.

Para realizar o que pretende, um STI inclui em sua estrutura básica, conhecimentos especializados para tratar questões, tais como: O que ensinar? A quem ensinar? Como ensinar? Isto remete diretamente, respectivamente, aos domínios de estudos: Inteligência Artificial (IA), Psicologia Cognitiva e a Pedagogia (KEARSLEY, 1987).

A arquitetura tradicional de um STI conta com quatro componentes principais: modelo do domínio, modelo do estudante, modelo de tutoria e modelo de comunicação. Cada um destes componentes pode ser descrito como segue:

Modelo do domínio (ou modelo do Especialista): representa do domínio de conhecimento, assumido como aquele conhecimento que se pretende que o estudante adquira, normalmente incluindo conceitos, relacionamentos entre conceitos;

Modelo do estudante: representa aspectos relevantes do conhecimento do estudante sobre o domínio, normalmente determinado pelas soluções dos estudantes aos problemas ou outras interações com o sistema, por exemplo: desempenho em fórum de discussão, sendo dinamicamente atualizado. Contudo, além desta parcela de informação do estudante com respeito ao domínio de conhecimento, em abordagens mais recentes, tal modelo se expande para incluir características independentes de domínio, oriundas, por exemplo, de sociabilidade, de estados afetivos e motivacionais. Mais detalhes, com um apanhado abrangente e significativo do que tem ocorrido no assunto, podem ser obtidos em (CHRYSAFIADI & VIRVOU, 2013);

Modelo de tutoria (ou modelo pedagógico): representa o conhecimento sobre estratégias pedagógicas, permitindo ao sistema individualizar suas ações: o sistema interpreta o conhecimento do estudante do modelo do estudante, comparando-o ao modelo de domínio, para assim tomar diferentes tipos de decisão;

Modelo de comunicação: diz respeito à interface que permite a interação com o estudante.

Finalmente, ressalta-se que ao longo do tempo essa arquitetura vem

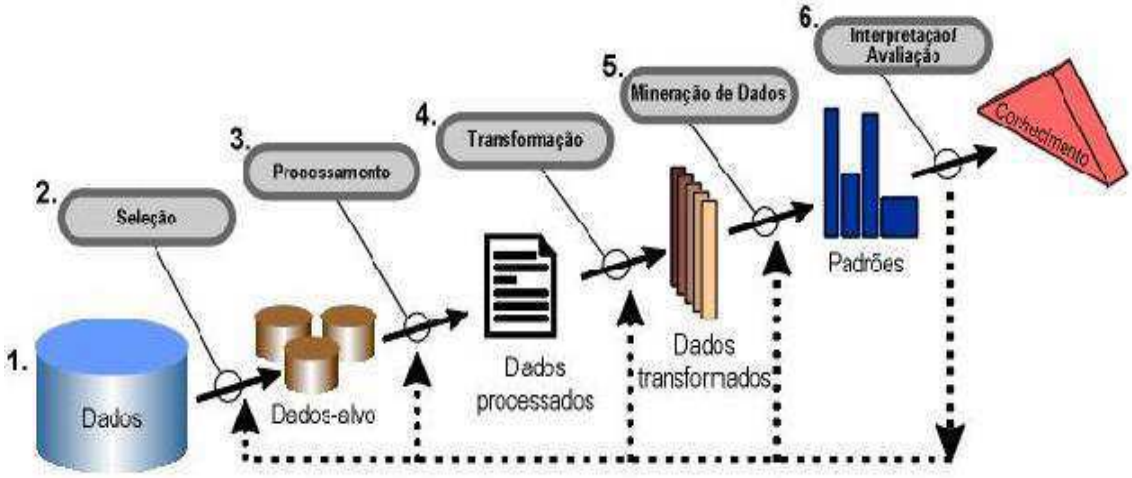
sofrendo vários incrementos em cada um dos seus módulos centrais, considerando principalmente os avanços em cada uma das áreas que lhe dão suporte. Assim, tem havido significativas evoluções nos modelos do domínio, do estudante, pedagógico e de comunicação. Além disso, um dos avanços relevantes na concepção de STI tem sido a adoção de uma abordagem de agentes inteligentes (COSTA et al., 1995), tanto do ponto de vista da Inteligência Artificial, quando da Engenharia de Software (COSTA et al., 2002). Isso tem permitido uma significativa evolução, inclusive influenciando no que se pretende no momento que é uma integração efetiva entre o que propõe um AVA e a ambiciosa proposta de um STI com respeito a adaptação e personalização (JAIN et al., 2015). Ademais, há ainda uma tendência de STI orientado a dados (KOEDINGER et al., 2013), onde se busca integrar a esse tipo de sistema, técnicas de mineração de dados educacionais e *learning analytics* (LIN & PEREZ, 2015; GAŠEVIĆ et. al., 2015; PAPAMITSIOU & ECONOMIDES, 2014). A presente proposta visa contribuir com essa tendência de emprego de técnicas de Inteligência Artificial em Educação (WOOLF, 2013; LIN & PEREZ, 2015; KOTSIANTIS, 2009), por exemplo, apontando e contribuindo cada vez mais para uma aprendizagem personalizada (O'DONNELL et al., 2015).

2.3 Descoberta de Conhecimento em bases de dados

Nas últimas décadas, com a ampliação dos meios e facilidades para coleta e armazenamento de dados, permitindo a produção de grandes volumes de dados, constatou-se diversos avanços em sistemas de informação e apoio a decisão, notadamente naqueles que têm sido denominados de sistema de apoio a decisão baseado em dados, do inglês *data-driven decision making*. O crescimento no volume de dados, entretanto, potencialmente provoca maiores dificuldades nas tarefas de extração de informação útil para apoiar o processo de decisão. Ao mesmo tempo, esses dados representam oportunidades para descoberta de conhecimento relevante, possibilitando importantes decisões e planejamento mais adequado. Nesse contexto, para abordar as referidas dificuldades, surgiu um campo de pesquisa interdisciplinar com a abordagem

denominada Descoberta de Conhecimento em Bases de Dados, de KDD sigla inglesa para Knowledge-Discovery in Databases e, uma de suas etapas particulares, a Mineração de Dados, beneficiando-se de áreas, tais como: Estatística, aprendizagem de máquina, reconhecimento de padrões, banco de dados. KDD pode ser visto como um termo utilizado por pesquisadores de Inteligência Artificial para o processo de encontrar conhecimento em dados (Aprendizagem de Máquina), sendo a mineração de dados uma parte integrante da descoberta de conhecimento em banco de dados. Portanto, a Mineração de Dados (MD, do inglês, Data Mining, DM), pode vista como uma etapa principal do KDD. Em KDD verifica-se ainda a inclusão de mais duas grandes etapas: pré-processamento de dados (preparação de dados, abrangendo mecanismos para captação, organização e tratamento dos dados) e pós-processamento dos resultados obtidos na mineração de dados. Neste sentido, de acordo com (FAYYAD et al.al. 1996), “KDD é um processo, não trivial, de identificação de padrões, a partir de dados, que sejam válidos, novos, potencialmente úteis e compreensíveis”. Trata-se, portanto, de uma definição abrangente, na qual KDD é descrito como um processo geral de descoberta de conhecimento composto pelas três grandes etapas mencionadas. Assim, os padrões mencionados deverão trazer algum benefício novo que possa ser compreendido rapidamente pelo usuário para uma possível tomada de decisão. Na Figura 2.3, mostra-se, de maneira simplificada, o processo KDD (TAN et al., 2009), apresentando as cinco etapas que constituem o processo

Figura 2.3-Etapas do processo KDD



Fonte: FAYYAD; PIATETSKY-SHAPIRO; SMYTH, 1996.

Inicialmente, por meio de um entendimento bem definido do domínio da aplicação, é necessário selecionar as bases de dados, bem como os dados, que serão usados no processo de descoberta de conhecimento. Em seguida, é efetuada a limpeza e o pré-processamento, uma vez que, frequentemente, os dados são encontrados com inúmeras inconsistências. Essas tarefas são fundamentais, pois o objetivo é eliminar incongruências, de modo que não influenciem o resultado dos algoritmos de mineração que serão aplicados. Posteriormente, realiza-se a transformação que consiste em reduzir ou projetar tais dados.

Essas três etapas iniciais podem ser agrupadas, originando uma grande fase nesse processo, conhecida como preparação de dados. A partir da preparação apropriada dos dados, tendo-se os dados pré-processados, inicia-se a etapa de mineração que, conforme já mencionado, consiste em escolher técnicas e algoritmos que possibilitem a extração de padrões.

Finalmente, efetua-se a etapa de avaliação que compreende na interpretação dos padrões minerados, na qual os resultados obtidos são validados. Após uma análise minuciosa, usa-se o conhecimento diretamente, incorporando-o a sistemas de apoio a decisões, ou simplesmente se documenta esse conhecimento, expondo-o às partes interessadas. Portanto, essa fase é importante, pois assegura que apenas resultados úteis e válidos sejam utilizados.

Há uma falta de consenso entre os autores sobre uma definição para o termo Mineração de Dados, dificultando a consolidação de uma definição única, o que faz muitos pesquisadores optarem por considerar o termo mineração de dados como um sinônimo de KDD (HAN et al., 2011). Nesta pesquisa, após esse esclarecimento inicial, os termos KDD e Mineração de Dados, por simplicidade e já tendo feitos as considerações técnicas, não serão mais distinguidos. Assim, o processo de mineração de dados passará a ser visto como consistindo de três passos executados em sequência:Pré-processamento de dados, análise de dados e a interpretação resultante, na qual o modelo selecionado é testado sobre novos dados para ratificar ou não o resultado esperado.

2.3.1 Mineração de Dados

Antes de discutir a etapa de Mineração de Dados (do inglês, *Data Mining*, DM) em si, convém comentar a ampla etapa de pré-processamento, ressaltando desde a coleta de dados até a seleção de atributos e balanceamentos de dados. Assim, salienta-se, que as técnicas de pré-processamento de dados são frequentemente utilizadas para melhorar a qualidade dos dados, visando, por exemplo, um uso mais apropriado dos algoritmos de mineração.

Analisar dados que não tenham sido pré-processados cuidadosamente pode produzir resultados errôneos. As tarefas principais envolvidas nesta etapa de pré-processamento de dados, incluem limpeza dos dados, integração de dados, dados desbalanceados, transformação de dados, redução de dimensionalidade e discretização dos dados. Na limpeza dos dados, objetiva-se tratar dados ruidosos e dados faltantes ou incompletos. Na integração de dados, integra-se dados de múltiplas fontes em uma única, resolvendo problemas de inconsistências e redundância de dados. Na transformação de dados, os dados são transformados para se adequar ao formato requerido pelos algoritmos de mineração de dados. A tarefa de redução de dimensionalidade ou, particularmente seleção de atributos, é muito importante quando se trabalha com conjunto de dados com alta dimensionalidade, ou seja,

com grande número de atributos, o que aumenta o custo computacional de várias técnicas de mineração de dados. Em classificação, por exemplo, comumente, no caso de seleção de atributos, trabalha-se apenas com um subconjunto, escolhido, do conjunto de atributos original. A escolha pode ser manual, por um especialista no domínio de aplicação dos dados, ou automática, por algum algoritmo de seleção automática de atributos. Na discretização dos dados, valores numéricos são transformados em categóricos, ou vice-versa, ou ainda alguma transformação de numérico em numérico.

2.3.1.1 Tarefas de Mineração de Dados

Para descobrir conhecimento que seja relevante, é importante estabelecer metas bem definidas. Segundo (FAYYAD et al,1996), no processo de descoberta de conhecimento as metas são definidas em função dos objetivos na utilização do sistema, podendo ser de dois tipos básicos: verificação ou descoberta. Quando a meta é do tipo verificação, o sistema está limitado a verificar hipóteses definidas pelo usuário, enquanto que na descoberta o sistema encontra novos padrões de forma autônoma. A meta do tipo descoberta, em geral, está relacionada com as seguintes tarefas de mineração de dados com as categorias: métodos de predição e métodos de descrição.

Tarefas Preditivas objetivam fazer predição acerca de valores de dados usando resultados conhecidos de outros dados, isto é, predizer o valor de um determinado atributo (variável) baseado nos valores de outros atributos. O atributo a ser predito é comumente conhecido como a variável preditiva, dependente ou alvo, enquanto que os atributos usados para fazer a predição são conhecidos com as variáveis preditoras, independentes ou explicativas. De modo mais abstrato, a predição se utiliza de uma tupla de variáveis para predizer outras variáveis ou valores desconhecidos (FAYYAD et al. 1996).

Tarefas Descritivas procuram encontrar padrões (correlações, tendências, grupos, trajetórias e anomalias) que descrevam os dados.

As metas de predição e descrição são alcançadas abordando alguma das seguintes tarefas e associadas técnicas de mineração de dados: classificação,

regressão, agrupamento, regras de associação. Entre as tarefas preditivas, incluem-se aquelas que usam técnicas de classificação ou regressão. Entre as tarefas de descrição incluem-se: agrupamento e regras de associação.

Diferentes estratégias podem ser utilizadas para minerar as bases de dados na busca por indícios que possam relacionar dados ou fatos. As principais estratégias empregadas, nesta tarefa, incluem técnicas para: classificação e regressão, como parte do aprendizado supervisionado, agrupamento e regras de associação, como parte do aprendizado não supervisionado. Em todas as estratégias, o objetivo maior é o de poder generalizar o conhecimento adquirido para novas ocorrências do fenômeno ou para outros contextos ou situações parecidas com a utilizada na construção do modelo computacional. Em cada uma destas estratégias diferentes técnicas e algoritmos podem ser aplicados.

Classificação é uma operação que consiste na busca por um modelo de algum tipo que possa ser aplicado a dados não classificados visando categorizá-los em classes. Um objeto é examinado e classificado de acordo com uma classe definida (HARRISON,1998). A construção do modelo segundo esta estratégia, pressupõe o conhecimento prévio das possíveis classes e a correta classificação dos exemplos usados na modelagem. Em síntese, busca-se a aprendizagem de uma função que mapeia um item de dado em uma de várias classes pré-definidas.

Regressão consiste na busca por uma função que represente, de forma aproximada, o comportamento apresentado pelo fenômeno em estudo. Ou seja, é aprender uma função que mapeia um item de dado para uma variável de predição real estimada (FAYYAD at al., 1996).

Agrupamento (do inglês, clustering) é um processo de partição de uma população heterogênea em vários subgrupos ou clusters de acordo com alguns critérios de homogeneidade (HARRISON, 1998). Isto é, processo de identificação de grupos de dados dotados de características similares com os do mesmo grupo e onde os grupos tenham características diferentes entre si. A principal diferença entre esta abordagem e a classificação é que no agrupamento não se tem conhecimento prévio sobre as classes predefinidas, os registros são agrupados de acordo com a semelhança.

Associação se baseia em identificar fatos que possam ser direta ou indiretamente associados (HARRISON, 1998).

2.3.1.2 Modelos e Algoritmos de Predição em Mineração de dados

A seguir são descritos sucintamente alguns modelos e algoritmos preditivos, buscando uma cobertura significativa em relação ao viés de método de aprendizagem considerado, assim abrangendo: métodos baseados em busca (ex.: indução de árvore de decisão), métodos baseados em otimização (ex.: máquina de vetores de suporte (SVM)), métodos probabilísticos (ex.: Naive Bayes), métodos baseados em distâncias (ex.: k vizinho mais próximo (KNN)) (FACELI et al. 2011). Dentre tais métodos, são ainda distinguidos os que se realizam via algoritmos White box ou Black box, sendo os White box aqueles cujos modelos são assumidos como compreensíveis pelos usuários, ao passo que os black box são de compreensibilidade difícil. Ressalta-se que os métodos estão apenas comentados resumidamente, mas com as devidas referências para onde estão bem explicados, pois são métodos e técnicas que são bem esclarecidas e detalhadas em diversos livros de aprendizagem de máquina e mineração de dados, a exemplo de (MICHEL, 1997), (TAN et al. 2006), (WITTEN & FRANK, 1999).

MÉTODOS BASEADOS EM BUSCA

Nesta categoria, um problema de aprendizado de máquina é formulado como um problema de busca em um espaço de estados com possíveis soluções. Assim, há vários métodos nesta categoria, incluindo-se os que geram como modelo árvore de decisão ou regras de classificação.

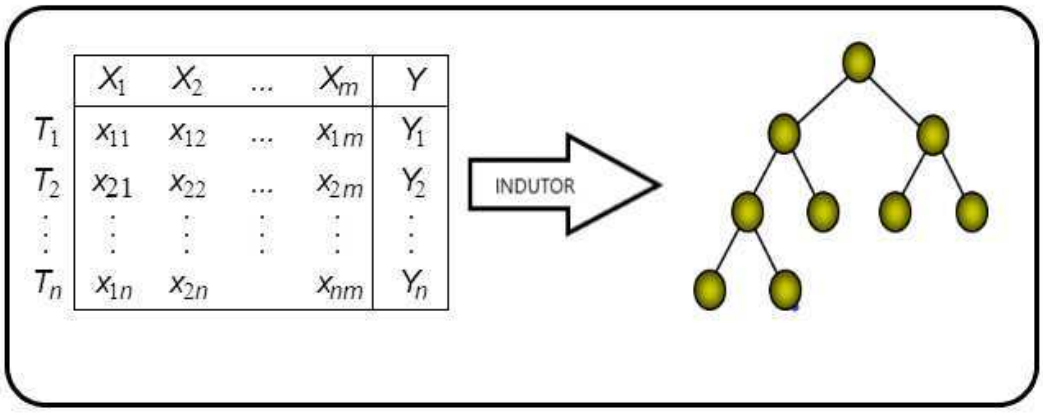
Árvore de Decisão

Indução de Árvore de Decisão é uma técnica de aprendizagem supervisionada, consistindo de uma coleção de nós internos e externos (folhas) conectados por ramos, organizados em uma estrutura hierárquica, refletindo a ideia de uma árvore invertida, a qual se desenvolve da raiz para as folhas.

Essa estrutura hierárquica pode ser interpretada como uma progressão da análise de dados no sentido de desempenhar uma tarefa de predição/classificação. Em cada nível da árvore tomam-se decisões acerca da estrutura do nível seguinte até atingir os nós terminais (nós folha) (BARANAUSKAS & MONARD, 2000), funcionando como um tipo de questionário. Ou seja, com tais técnicas geram-se uma estrutura representativa de uma sequência de decisões por meio das quais ocorrem sucessivas divisões em um conjunto de dados inicial (nó raiz) até que o mesmo seja representado por diversas classes dentro dos nós filhos gerados (Breiman et al.,1984). Quando nenhuma outra subdivisão dos dados é possível, os subconjuntos finais são denominados nós terminais ou folhas.

Basicamente, um algoritmo para construção de uma árvore de decisão recebe como entrada um conjunto de exemplos rotulados, contendo pares atributo e valor, gerando uma saída com a referida estrutura de dados hierárquica, isto é, a árvore de decisão, conforme ilustrado na Figura 2.4. Assim, um exemplo é descrito pelos valores dos atributos e pelo predicado meta (atributo classificador). O valor do predicado meta é chamado classificação do exemplo. Para cada um dos possíveis valores de atributos, tem-se ramo para outra árvore de decisão (sub-árvore). Cada sub-árvore contém a mesma estrutura de uma árvore.

Figura 2.4- Ilustração do processo geral de construção de uma árvore de decisão

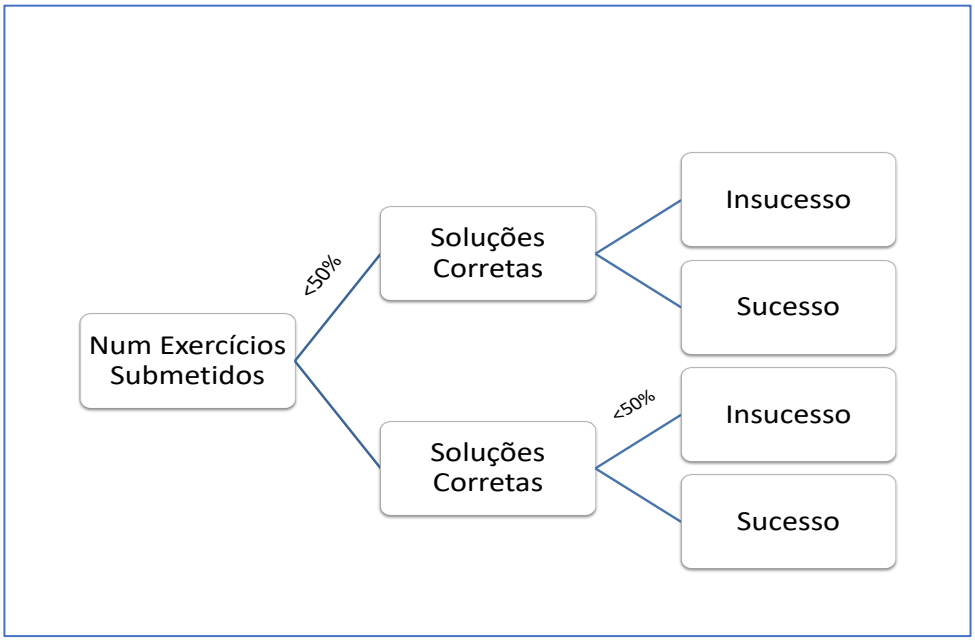


Fonte: Adaptado de (BARANAUSKAS & MONARD, 2000)

Uma árvore de decisão possui correspondência direta com um conjunto

de regras de classificação. Cada caminho da raiz até uma folha representa uma destas regras. Cada percurso da árvore de decisão, desde um nó raiz até um nó folha é convertido em uma regra, onde a classe do nó folha corresponde à classe prevista pelo consequente da regra e as condições ao longo do caminho correspondem às condições do antecedente da regra. A aplicação da AD é realizada levando em conta três elementos principais: um conjunto de questões que delimita a divisão dos dados, um critério para estabelecer a melhor divisão na obtenção de nós filhos e uma regra de parada para as subdivisões (stop-splitting rule) (BARANAUSKAS; MONARD, 2000). A Figura 2.5 mostra um exemplo simplificado de uma árvore de decisão para diagnóstico de estudantes, quanto ao seu sucesso ou insucesso em uma determinada disciplina.

Figura 2.5- Exemplo de Árvore de decisão



Fonte: Autora

Há diferentes abordagens para construção de árvores de decisão, tendo vários algoritmos para implementá-las. Um deles é o algoritmo de Hunt, que é a base de muitos algoritmos de indução de árvores de decisão existentes na literatura, incluindo o ID3, C4.5 (QUINLAN, 1993), C5.0 proposto por

QUINLAN, e CART (Classification and Regression Trees, proposto por Breiman et al. (1984). Além desses, há outras abordagens para árvores de decisão, a exemplo de ADTree, RandomTree e REPTree (WITTEN E FRANK, 2011), os quais foram utilizados na presente pesquisa.

Regras de classificação

Os classificadores baseados em regras utilizam-se de técnicas para classificar registros, usando um conjunto de regras no já bem conhecido formato SE antecedente ENTÃO consequente — em que o consequente é composto de apenas um par “atributo = valor”, sendo o atributo alvo de classificação (TAN et al., 2009). No antecedente, a condição consiste em um ou mais testes de atributos (por exemplo, idade = jovem e gênero = feminino e estudante = estudioso). Nesse caso, o consequente da regra contém uma previsão da classe (nesse caso, estamos prevendo se estudante não vai ser reprovado). Portanto, considera-se o seguinte modelo de regra: Regras de classificação na forma SE <condição> ENTÃO <classificação>, cuja interpretação é “se os valores assumidos pelos atributos de um registro do conjunto de treinamento satisfazem as condições do antecedente da regra, então o registro recebe a classe indicada pelo valor do atributo de classificação”.

Dentre vários algoritmos que produzem regras de classificação a partir de dados, induzindo conhecimento como regras proposicionais, baseados em pares <atributo, valor>, destacam-se: PRISM (CENDROWSKA, 1987), RIPPER (COHEN, 1995), OneR (WITTEN & FRANK, 1999), dentre outros. Na biblioteca WEKA, o RIPPER consta como JRip, além do que no presente trabalho se utilizou também dos seguintes algoritmos implementados no WEKA: NNge, OneR, Prism e Ridor, os quais foram usados nos estudos exploratórios constantes nos Capítulos 4 e 5 deste documento de tese.

MÉTODOS BASEADOS EM OTIMIZAÇÃO

Estes métodos podem ser caracterizados por buscarem a hipótese que descreve os dados recorrendo-se à otimização de alguma função, ou seja, um

problema de aprendizado é formulado como um problema de otimização, o qual consiste em minimizar ou maximizar uma determinada função objetivo. Nesta abordagem incluem-se as técnicas: Máquinas de vetores de suporte e as redes neurais artificiais, as quais estão sucintamente descritas a seguir.

Máquina de Vetores de Suporte

Máquina de Vetores de Suporte (do inglês, Support Vector Machine (SVM)) é um método baseado na teoria de aprendizagem estatística e otimização matemática (VAPNIK, 1995). Deste modo, constitui um algoritmo supervisionado utilizado para a tarefa de classificação que utiliza um hiperplano como separador de classes (TAN et al. 2005). Este hiperplano é descoberto usando os vetores de suporte (conjunto de treinamento) e funciona como um suporte para o limite da decisão ao classificar. Com o SVM resolve-se tanto problemas de classificação quanto de regressão, envolvendo duas classes, mas pode ser estendido para problemas multi-classes. Foi proposto por Vapnick e é uma abordagem de aprendizagem supervisionada baseada na noção de kernel, ou mais especificamente, de funções denominadas kernels. Entre as funções kernel mais utilizadas, incluem-se: polinomial (incluindo o kernel linear), os de função de base radial (radial basis function – RBF) e os sigmoidais. Cada um deles tem parâmetros em suas respectivas funções, tal como ilustrados na Tabela 1. Tais parâmetros precisam ser determinados pelos usuários. Em se tratando de kernels polinomiais, quando o parâmetro d assumir valor igual 1, tem-se um kernel linear.

Tabela1 Algumas Funções Kernel

Tipo de Kernel	**Função $K(x_i,x_j)$**	**Parâmetros**
Polinomial	$(\delta(x_i \cdot x_j)+\kappa)^d$	δ, κ e d
Gaussiano (RBF)	$\exp(-\sigma\|x_i - x_j\|^2)$	σ
Sigmoidal	$\tanh(\delta(x_i \cdot x_j)+\kappa)$	δ e κ

Fonte: Adaptado de (FACELI et al. 2011)

Redes Neurais Artificiais

Conforme (HAYKIN, 2000), as redes neurais ou redes neurais artificiais (RNA) representam uma tecnologia que tem raízes em muitas disciplinas: neurociência, matemática, estatística, física, ciência da computação e engenharia, consistindo em um método para solucionar problemas da área de inteligência artificial, através da construção de um sistema que tenha circuitos que simulem o cérebro humano, inclusive seu comportamento, ou seja, aprendendo, errando e fazendo descobertas. São técnicas computacionais que apresentam um modelo inspirado na estrutura neural dos organismos inteligentes, que adquirem conhecimento através da experiência (GOEBEL e GRUENWALD, 1999).

Há diferentes abordagens para as redes neurais artificiais, iniciando-se com a rede perceptron, desenvolvida na década de 50, resolvendo apenas problemas simples, linearmente separáveis. No entanto, para resolver problemas não linearmente separáveis, a proposta do perceptron foi evoluída para as redes do tipo perceptron multicamadas (do inglês, MLP – Multilayer Perceptron), apresentando uma ou mais camadas intermediárias de neurônios e uma camada de saída. A MLP e outras abordagens estão descritas em detalhes em (HAYKIN, 2000) e uma descrição mais sucinta pode ser encontra em (FACELI et al. 2011). Nesta pesquisa, fez-se uso de MLP.

MÉTODOS PROBABILÍSTICOS

Nesta categoria, inclui-se a classificação Bayesiana via Naive Bayes. Trata-se de um método probabilístico, utilizado na tarefa de classificação em aprendizado supervisionado, baseado no conhecido teorema de Bayes (HAN et al., 2011). Segundo esse teorema, é possível encontrar a probabilidade de um certo evento ocorrer, dada a probabilidade de um outro evento que já ocorreu, conforme a fórmula a seguir (TAN et al., 2009):

Dados A e B como eventos:

- P(A) e P(B) são as probabilidades de A e B ocorrerem, sem levar em

conta um no outro;

- P(A|B) é a probabilidade condicional, é a probabilidade de A dado que B é verdadeira; e
- P(B|A) é a probabilidade de B dado que A é verdadeira. Estudos comparativos mostraram que os algoritmos Bayesianos, chamados de Naïve Bayes, obtiveram resultados compatíveis com os métodos de árvore de decisão

$$P\ (A/B) = \frac{P\ (A).P(B/A)}{P(B)}. \quad (1)$$

O classificador Naive Bayes é denominado ingênuo (naive) por assumir que atributos são condicionalmente independentes para um determinado atributo classificador. Apesar dessa premissa "ingênua" e simplista, o classificador se comporta bem em várias tarefas de classificação (MITCHELL, 1997). Outro fator que torna o Naive Bayes eficiente é que ele realiza a leitura dos dados do conjunto de treinamento apenas uma vez, para com isso estimar todas as probabilidades requeridas na classificação. O modelo pode ser usado de forma incremental, além do fato de poder ser alterado facilmente com a inclusão de novos dados uma vez que a probabilidade pode ser facilmente recalculada.

Existem dois tipos de modelos estatísticos para o classificador Naive Bayes, o modelo binário e o modelo multinomial. O modelo binário representa um documento através de um vetor binário, indicando a não-ocorrência de um atributo com o valor 0 (zero), enquanto o valor 1 (um) representa no mínimo uma ocorrência. Já o modelo multinomial assume que o documento é representado por um vetor de valores inteiros, caracterizando o número de vezes que cada termo ocorre no documento (MITCHELL, 1997).

MÉTODOS BASEADOS EM DISTÂNCIAS

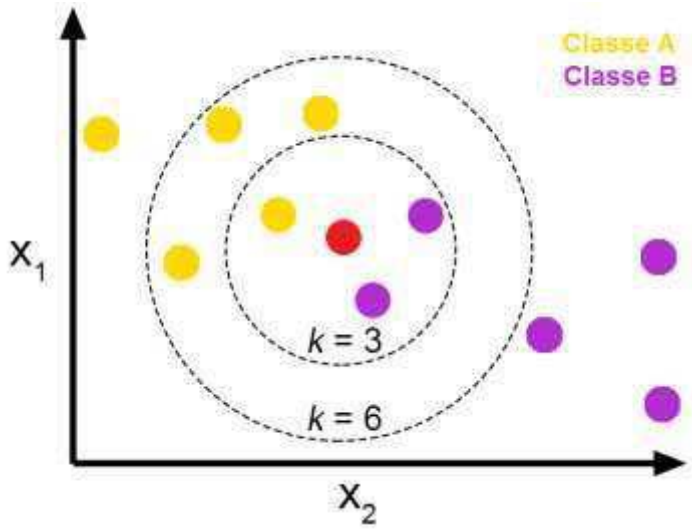
Nessa categoria, incluem-se os classificadores de vizinho mais próximo, trazendo uma proposta diferente, não-gulosa. Esse tipo de classificador

envolve um algoritmo simples, que armazena todos os casos disponíveis e classifica novos casos com base em uma medida de similaridade (por exemplo, funções de distância) aos casos já armazenados.

O kNN (k Nearest Neighbors) é o principal algoritmo desse tipo de classificador e foi descrito pela primeira vez no início de 1950. O método pode consumir muito tempo de processamento, dependendo da quantidade de exemplos do conjunto de treinamento, e não ganhou popularidade até os anos 1970, quando o aumento do poder de computação se tornou disponível. Desde então, tem sido amplamente utilizado na área de reconhecimento de padrões (HAN et al., 2011). A Figura 2.6 ilustra o processo de classificação, usando o kNN.

A definição do valor de k (número de vizinhos próximos) influencia no processo de classificação de novos dados. No exemplo ilustrado, se k=3, o novo elemento seria classificado na classe B, pois teria 2 vizinhos mais próximos (maioria) nessa classe. Para k=6, o elemento seria classificado na classe A, com 4 vizinhos mais próximos.

Figura 2.6- Exemplo da classificação do kNN para dois valores de k



Fonte: (HAN et al., 2011).

2.3.1.3 Técnicas de Seleção de Atributos

O objetivo principal da seleção de atributos é escolher um subconjunto de variáveis de entrada, eliminando as características ou atributos considerados irrelevantes ou sem informação para efeitos da tarefa preditiva em apreço. De fato, várias aplicações reais apresentam um grande número de atributos (FACELI et al, 2011), isso pode acarretar problemas de dimensionalidade, parte destes atributos podem ser irrelevantes, ou redundante, gerando ruídos. Para lidar com esses problemas, existem as técnicas de redução de dimensionalidade, as quais podem ser divididas em duas categorias: agregação e seleção de atributos (FACELI et al, 2011). Na categoria agregação, as técnicas utilizadas substituem os atributos originais por novos atributos formados pela combinação de grupos de atributos, tendo a técnica de Análise de Componentes Principais, do inglês: PCA, Principal Component Analysis, como uma das bem conhecidas, além de uma extensão sua, a Kernel PCA. Já a categoria seleção de atributos, adotada no presente trabalho, é constituída por técnicas que mantêm uma parte dos atributos originais e descartam os demais.

As técnicas de seleção de atributos ajudam a lidar com esses tipos de problemas, implementando operações que tomam como entrada um conjunto de atributos A e produz na saída um subconjunto de A selecionado.

Além dos algoritmos de classificação descritos anteriormente, é importante mencionar aqui que existem vários métodos de seleção de atributos, basicamente divididos em três categorias (FACELI et al, 2011):

Embutida: a seleção do subconjunto de atributos é embutida ou integrada no próprio algoritmo de aprendizado.

Abordagem independente de modelo (Baseada em **filtro**): realizada no pré-processamento, filtrando do conjunto original um subconjunto de atributos, portanto sem levar em consideração o algoritmo de mineração de dados que será aplicado aos atributos selecionados, ou subconjunto.

Abordagem dependente de modelo (Baseada em ***wrapper***): utiliza o

próprio algoritmo de aprendizado como uma caixa preta para a seleção. Para cada possível subconjunto, o algoritmo é consultado e o subconjunto que apresentar a melhor combinação entre redução de taxa de erro e redução do número de atributos é em geral selecionado.

Entre os algoritmos utilizados nestas categorias, considerando-se suas implementações na biblioteca WEKA, incluem-se: Correlation-based Attribute evaluation (CB), Chi-Square Attribute evaluation (CH), Gain-Ratio Attribute evaluation (GR), Information-Gain Attribute evaluation(IG), Relief Attribute evaluation (RF). Ressalta-se que esses são exemplos de implementação do método filtro.

Uma outra maneira de categorizar os métodos de seleção de atributos é a de considerá-los em 2 dois tipos, considerando-se o propósito de ordenação, avaliando e mensurando a relevância dos atributos. Assim, tem-se as categorias de técnicas:

a. Attribute Subset Evaluators (Avaliadores de Subconjunto de Atributos) – Neste caso, utilizam-se um subconjunto de atributos e retornam uma medida numérica que orienta a busca.
b. Single-Attribute Evaluators (Avaliadores de Atributo Único) – Neste caso, utiliza-se um método de pesquisa para ordenação (ranqueamento), produzindo uma lista de atributos, conforme suas relevâncias, podendo descartar alguns atributos.

2.3.1.4 Construção e Avaliação de Modelos Preditivos

Na Seção 2.3.1.2., descreveu-se alguns modelos preditivos utilizados em mineração de dados. No entanto, não foi discutido ainda o processo de construção desses modelos, nem tampouco técnicas para avaliá-los. Assim, menciona-se aqui métodos para treinamento, validação e teste, incluindo-se métodos de amostragem, incluindo-se holdout, validação cruzada e bootstrap (FACELI et al, 2011). No presente trabalho se investiu no método de validação cruzada (k-fold cross-validation), no qual o conjunto de exemplos é dividido em k subconjuntos de tamanhos aproximadamente iguais. Assim, os objetos de k-1 partições são utilizados no treinamento do preditor, o qual é então testado na partição restante (FACELI et al, 2011). Há variações de uso desse método, mas isso não vai ser discutido aqui, porém nesta referência há mais detalhes a respeito dela.

Com respeito a medidas de desempenho, algumas das métricas comumente usadas, inclusive no presente trabalho, são as seguintes:

(i) Taxa de acerto ou acurácia total (*Accuracy*): calculada pela soma dos valores da diagonal principal da matriz de confusão, dividida pela soma de todos os elementos da matriz.

$$Accurácia\ (ACC) = \frac{\sum True\ Positive + \sum True\ Negative}{\sum Total\ Population} \quad . \qquad (2)$$

(ii) Taxa de Verdadeiros Positivos (*True Positive Rate*): Calcula-se a soma dos verdadeiros positivos sobre os verdadeiros.

$$TP\ rate = \frac{\sum True\ Positive}{\sum True\ Positive + \sum False\ Negative} . \qquad (3)$$

(iii) Taxa de Verdadeiros Negativos (*True Negative Rate*): Calcula-se a soma dos verdadeiros negativos sobre os negativos.

$$TN\ rate = \frac{\sum True\ Negative}{\sum True\ Negative + \sum False\ Positive} \quad . \qquad (4)$$

(iv) Média Geométrica (*Geometric Mean*): indica o equilíbrio entre duas medidas de classificação. Representa uma medida de trade-off comumente usada com conjuntos de dados desequilibrados e calculada como:

$$GM\ (Geometric\ Mean) = \sqrt{TP\ rate\ .\ TN\ rate}\quad . \tag{5}$$

Uma outra métrica, inclusive usada nesta pesquisa, é a f-measure, a qual é definida pela média harmônica entre Precisão (Precision) e Sensibilidade (Recall). Precisão é calculada pela proporção de exemplos positivos classificados corretamente entre todos aqueles preditos como positivos pelo preditor. Já o Recall corresponde à taxa de acerto na classe positive, sendo também chamada de taxa de verdadeiros positivos, tal como já mencionado anteriormente. Mais informação sobre essas métricas pode ser encontrada em *(FACELI et al, 2011).*

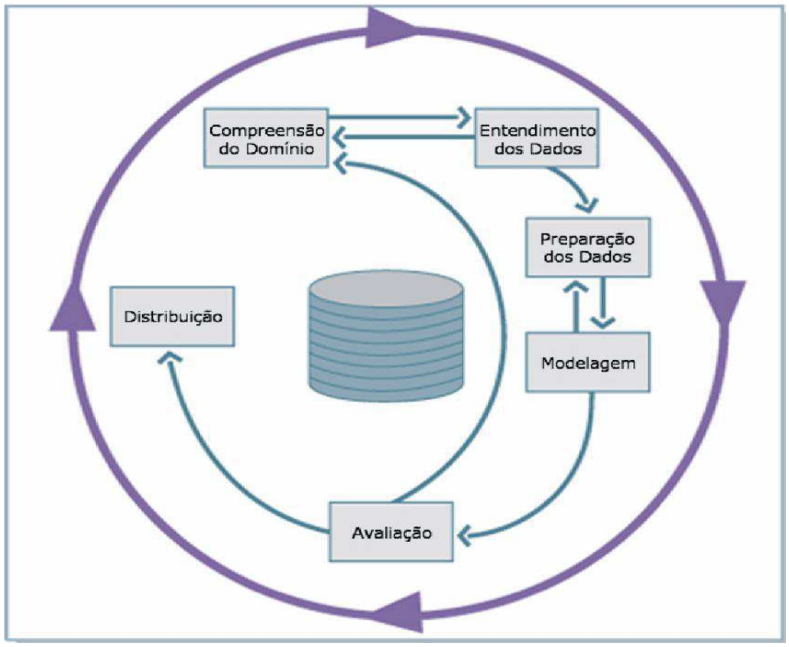
2.3.1.5 Ambiente Computacional para Mineração de Dados: WEKA

O ambiente de Waikato para análise de conhecimento (WITTEN, 1999), WEKA ('Waikato Environment for Knowledge Analysis'), conta com uma ampla biblioteca contendo implementação de algoritmos de mineração de dados. Tal biblioteca foi desenvolvida por pesquisadores do Departamento de Computação da Universidade de Waikato da Nova Zelândia (WITTEN et al, 2000). WEKA foi implementado em Java, cobrindo uma variedade de algoritmos, incluindo os que implementam árvores de decisão, regras de classificação, SVM, redes neurais artificiais, Naïve Bayes, KNN. A representação dos dados é feita em um formato específico e a mineração é feita através da leitura dos dados, obtidos de um arquivo previamente formatado: um arquivo no formato ARFF (Attribute-Relation File Format), um arquivo texto (ASCII) que descreve uma lista de instâncias compartilhando um conjunto de atributos, possuindo duas seções distintas: cabeçalho, que descreve os atributos e seus possíveis valores, e a seção de dados, a relação de instâncias.

2.3.2 Processo de Mineração de Dados e a Metodologia CRISP-DM

CRISP-DM é uma metodologia define uma sequência não rígida de etapas, buscando orientar o processo de construção e implementação de um modelo de mineração de dados, consistindo de uma sequência de seis etapas (CHAPMAN et al., 2000), desenvolvidas ciclicamente, tal qual na Figura 2.7, incluindo:

Figura 2.7-Metodologia CRISP-DM



Fonte: Adaptado de Chapman et al. (2000)

Compreensão do negócio (*Business understanding*): Foca na compreensão do objetivo do negócio ou problema a ser resolvido, formulando-o como um problema (domínio) de mineração de dados. Ressalta-se aqui o aspecto cíclico do diagrama, significando que a finalização de uma etapa, por exemplo a definição do problema, poderá ocorrer em mais de um ciclo.

Compreensão dos Dados (*Data understanding*): Inicia-se com uma coleção inicial de dados, verificando sua adequação para ajudar na solução buscada para o problema de negócio definido. Procede-se, em seguida, com atividades para buscar entendimento dos dados, visando descobrir os

primeiros insights sobre os eles, eventualmente identificando problemas de qualidade nesses dados.

Preparação dos Dados (*Data preparation*): abrange todas as atividades para construir o conjunto de dados final a partir dos dados brutos iniciais, incluindo a seleção de dados, a limpeza de dados, a construção de dados, a integração de dados e a formatação de dados.

Modelagem (*Modeling*): Visa-se construir algum tipo de modelo ou padrão que capture regularidade nos dados. Geralmente abrange a criação de vários modelos de mineração de dados, iniciando-se com a seleção de métodos de mineração de dados, prosseguindo-se com a criação de modelos e terminando com a avaliação de tais modelos.

Avaliação (Evaluation): avalia os modelos de mineração de dados criados na fase de modelagem, objetivando verificar se os modelos satisfazem os objetivos do negócio, estimando os resultados de forma rigorosa e obtendo confiança de que são válidos e confiáveis, antes de coloca-los em uso.

Entrega ou Implantação (Deployment): abrange as atividades para organizar conhecimento adquirido por meio de modelos de mineração de dados e apresenta-lo de forma que os usuários possam utilizá-lo na tomada de decisões.

2.3.3 Mineração de Dados Educacionais

A área emergente de Mineração de Dados Educacionais procura desenvolver ou adaptar métodos e algoritmos de mineração existentes, de tal modo que se prestem a compreender melhor os dados em contextos educacionais, produzidos principalmente por estudantes e professores, considerando os ambientes nos quais eles interagem, tais como AVAs, Sistemas Tutores Inteligentes (STIs), entre outros. Com tais métodos visa-se entender melhor o estudante no seu processo de aprendizagem, analisando-se sua interação com o ambiente. Assim, há a necessidade, por exemplo, de adequação dos algoritmos de mineração de dados existentes para lidar com especificidades inerentes aos dados educacionais, tais como a não independência estatística e

a hierarquia dos dados. Por outro lado, há uma necessidade significativa e urgente no provimento de ambientes computacionais apropriados para mineração de dados educacionais, oferecendo facilidades de uso para cada um dos atores envolvidos, notadamente ao professor.

As origens da pesquisa focada em Mineração de Dados Educacionais (EDM) podem ser localizadas em algumas iniciativas primeiras com workshops específicos dentro das conferências sobre Artificial Intelligence in Education (AIEd) e sobre Intelligent Tutoring Systems (ITS). Mas, foi somente em 2005, em Pittsburgh, EUA, que foi organizado o primeiro Workshop on Educational Data Mining, como parte do 20th National Conference on Artificial Intelligence (AAAI 2005). Daí em diante, houve mais algumas realizações deste workshop entre 2006 e 2007. Seguindo-se, em 2008 lança-se, em Montreal, Canadá, a primeira conferência em EDM: First International Conference on Educational Data Mining, evento que se estabeleceu e ganhou regularidade de realização anual. Em 2009, essa sociedade investiu na criação de um periódico e publicou o seu primeiro volume do JEDM - Journal of Educational Data Mining. Em 2011 constituiu-se a sociedade científica para EDM (International Educational Data Mining Society 2). Enfim, a área de EDM está bem consolidada internacionalmente, mas também nacionalmente, já tendo no Brasil, evento especializado no tema, a exemplo de workshop que ocorre no CBIE, Congresso Brasileiro de Informática na Educação.

2.4 Síntese

Neste capítulo apresentou-se uma descrição geral dos temas de apoio ao desenvolvimento da tese em apreço, servindo ainda para contextualizar adequadamente a pesquisa. Assim, foi dada uma descrição sobre alguns aspectos de educação online onde se verifica a problemática em um nível de abstração mais alto, na camada de aplicação, seguindo-se para o tema descoberta de conhecimento de conhecimento em bases de dados e sua particularização em mineração de dados educacionais, no qual se enfatizou tarefas preditivas, destacando-se técnicas e métodos envolvidos na presente pesquisa.

3 TRABALHOS RELACIONADOS

Neste capítulo, discutem-se alguns dos principais trabalhos relacionados à presente tese, considerando-se a proximidade à abordagem proposta, observada em três patamares de similaridade, primeiramente discutindo os mais fortemente relacionados, isto é, os que cobrem todos os aspectos da proposta, ou seja, obedecem um critério que contempla a predição antecipada, notadamente, aquela comprometida com confiabilidade e compreensibilidade do modelo, tal como expresso no Capítulo 1 desta tese, nos problemas norteadores e na declaração de objetivo geral e do Objetivo 5. Em seguida, vem uma outra parte um pouco menos similar, ainda endereçando o problema da predição antecipada, mas que não se refere à questão da compreensibilidade do modelo. Por fim, aparece um terceiro patamar com os que possuem similaridade em algum aspecto envolvido na abordagem proposta, por exemplo não focam predição antecipada, mas na predição realizada apenas próximo ao final do curso com os dados acumulados até esse momento, eventualmente com alguma proposta interessante na identificação de fatores, via seleção de atributos, influenciando no desempenho acadêmico de estudantes ou na interpretabilidade do modelo.

Considerando-se a maior ordem de importância enfatizada no parágrafo anterior, situando-se no primeiro patamar de similaridade, identifica-se em MARQUEZ-VERA et al. (2016) uma abordagem, em seu propósito geral, muito próxima à proposta nesta tese, pois a predição é realizada ao longo de diferentes períodos de tempo, distinguindo-se em passos temporais na busca pela predição antecipada, tendo ainda a preocupação com a compreensibilidade do modelo. Trata-se de um trabalho que pode ser visto como uma evolução de um anterior, (MARQUEZ-VERA et al., 2013), no qual já se aborda preliminarmente vários aspectos importantes envolvidos na abordagem da presente tese, mas, por exemplo, não investiu na questão da predição antecipada, tal como discutido mais a frente. Assim, nesta evolução, Marquez-Vera et al. (2016) propõe uma metodologia mais abrangente para predição, pretensamente tão cedo quando possível, de estudantes no ensino médio, mexicanos, propensos à evasão. Para tanto, usou uma vasta base de

dados, contendo atributos acadêmicos, sócio-econômicos e demográficos, os quais foram escolhidos por um processo adotado para seleção de atributos e utilizados pelos algoritmos preditivos, conforme o que se pretendia em cada passo no experimento, indo do passo Zero, momento inicial, até o passo VI, correspondendo a um momento do curso. Os algoritmos de classificação utilizados foram: Naive Bayes, SMO (SVM), IBk (KNN), JRip, J48 (C4.5), ICRM. Após experimentos, verificou-se a obtenção de resultados satisfatórios para predição antecipada entre as semanas 4 e 6. Já com respeito ao momento inicial, passo Zero, conseguiu-se bons resultados tanto com, quanto sem seleção de atributos, alcançando acurácia em torno de 86%. Como será visto na comparação com a abordagem aqui proposta, conforme capítulo 5, tais resultados têm uma boa correspondência com os obtidos nesta tese, mas ressalvando-se primeiramente que o problema abordado neste trabalho é sobre evasão, tendo como público alvo estudantes do ensino médio, dentre outras diferenças importantes a serem ressaltadas posteriormente.

Em MARQUEZ-VERA et al. (2013), propõe-se aplicar técnicas de mineração de dados para predição de reprovação e evasão em escolas de nível médio. Para tanto, aplicou-se métodos caixa-branca como indução de regras e de árvores de decisão, sobre dados, inclusive não-acadêmicos, sobre 670 estudantes do ensino médio, obtidos de 3 fontes: Questionário, notas dos alunos no ensino básico e notas dos alunos no ano atual durante o curso. Nesta proposta, realizou-se um pré-processamento de dados, envolvendo operações, tais como: limpeza, integração, discretização e transformação das variáveis. Além do mais, realizou-se ainda um estudo para identificar quais características ofereceriam melhores resultados, usando-se 10 algoritmos de seleção de atributos, presentes na biblioteca WEKA, identificando-se os atributos mais pontuados nos referidos algoritmos e ranqueando-os. Após essa fase de seleção de atributos, aplicou-se algoritmos de classificação na base de dados resultante, assim como na original, permitindo comparação. Trata-se de uma proposta interessante e bem similar à parte preditiva da presente proposta, porém não contempla o requisito da predição no tempo inicial e não contempla a predição temporal, dentre outras limitações. Além disso, o escopo é diferente da presente pesquisa, tendo ainda um custo mais alto em termos de

coleta de dados, por exemplo, no emprego de questionários. Para predição, utilizou dez algoritmos de classificação com validação cruzada para estimar desempenho final e antecipar o insucesso escolar de estudantes, similarmente aos que foram usados por Marquez-Vera et al.(2010). Assim, eles desenvolveram duas abordagens para resolver o problema de classificar dados desbalanceados mediante o reequilíbrio de dados e a utilização a classificação de custo sensível (ELKAN, 2001) com ambas as abordagens apresentando resultados satisfatórios.

A pesquisa reportada em (KHOBRAGADE & MAHADIK, 2015) segue uma abordagem similar à que foi discutida anteriormente, apresentando uma proposta para predição de insucesso de estudantes, investindo-se em algoritmos classificadores White-Box para induzir regras e árvore de decisão, particularmente envolvendo o uso de dois algoritmos para regras e dois para árvore de decisão. Além disso, utilizou ainda o algoritmo Naive Bayes. Em termos de pré-processamento, selecionou-se os 11 melhores atributos relacionados a dados reais sobre os estudantes, incluindo-se notas, antecedentes familiares, aspectos sociais, bem como seu desempenho acadêmico passado. O algoritmo Naive Bayes forneceu a melhor precisão com 87.12, portanto, um bom resultado. No entanto, comparando-se aos resultados na presente tese, nesta pesquisa não se investiu em predição antecipada, além do que, além de dados acadêmicos, utilizou-se de várias outras fontes de dados.

A abordagem aqui proposta estende resultados das pesquisas relatadas no exame de qualificação da autora desta tese em (DE ARAÚJO, 2016) e em COSTA et al. (2017), em parte aqui apresentadas no Capítulo 4, focalizando a questão da predição antecipada por meio de um estudo minucioso sobre a efetividade de alguns algoritmos preditivos sobre dados de estudantes em cursos de programação introdutória, tanto na modalidade presencial, quanto a distância. Assim, obteve-se resultados de predição, considerando-se o ajuste de parâmetros no pré-processamento, incluindo seleção de atributos e o fine tunning (ajuste fino), com o intuito de avaliar o ganho de desempenho dos algoritmos. Ademais, dentre outros aspectos, não se investia na predição em T_0, momento imediatamente antes de se iniciar o curso.

Uma outra pesquisa relacionada é reportada em ASIF et al. (2017), na qual se investe em predição acadêmica de estudantes relativamente ao final de um curso de graduação, pretendendo identificar o potencial de estudantes chegar até o quarto ano, tendo-se em consideração a predição antecipada com os indicadores acadêmicos obtidos antes de entrar no curso e os atributos de notas obtidos entre o primeiro e o segundo ano do curso. O algoritmo preditivo utilizado foi árvore de decisão, alegando obtenção de uma acurácia razoável. Comparado à presente abordagem, percebe-se, entre outros aspectos, um escopo diferente, não se restringindo ao propósito de realizar predição no âmbito de uma disciplina. Além disso, investiu-se apenas em um algoritmo de árvore de decisão, inclusive não reportando um bom resultado de acurácia.

Em MARBOUTI et al. (2016), investigou-se também o problema da predição antecipada, utilizando-se os algoritmos preditivos, tais como: árvore de decisão, SVM, *Redes neurais via Multilayer Perceptron*, Naive Bayes, Regressão Logística e k-NN. Assim, neste trabalho se usou dados do ano de 2013 para treinar os modelos e, em seguida, aplicou os modelos aprendidos em dados de 2014, considerando-se a acurácia como métrica para aferir o desempenho dos algoritmos. As predições foram obtidas experimentalmente com e sem Seleção de Atributos. Os algoritmos que mais se destacaram, em termos de acurácia, nesta abordagem apresentada de treino e teste foram k-NN e Regressão Logística. Um relevante ponto de observação que pode ser comparado com o presente trabalho é que nesta pesquisa foram utilizados como atributos para as predições, dados de desempenho dos estudantes sobre realizações em tarefas de casa, elaboradas e aplicadas pelos professores aos alunos, assim como equivalentemente aconteceu com as notas das atividades do Huxley utilizadas no presente trabalho. Não se discutiu explicabilidade dos modelos, nem se explicitou aspectos temporais da predição.

Na pesquisa reportada em BYDZOVSKA (2016), abordou-se o problema de prever as notas finais dos alunos no início do semestre com ênfase na identificação de alunos malsucedidos. Para isso, utilizou-se duas abordagens diferentes, sendo a primeira baseada em algoritmos de classificação e regressão. Essa abordagem foi considerada interessante quando utilizada para a predição de séries de cursos com um pequeno número de alunos. Os

algoritmos empregados foram: Máquina de Vetores de Suporte, Floresta Aleatória, Classificador baseado em regras, Árvore de Decisão, Part, IB1 e Naive Bayes. Neste estudo, o SVM alcançou o melhor desempenho. Os resultados foram melhorados, também usando dados sobre o comportamento social dos alunos nas previsões. A segunda abordagem usada foi em uma linha diferente, considerando técnicas de filtragem colaborativa e notas previstas, usando-se métodos baseados em distância, com base na similaridade das realizações dos alunos. Esta abordagem é distinta daquela discutida na presente abordagem. Na comparação mais geral, investiu-se em dados sobre o comportamento social dos estudantes, bem como não privilegiou algoritmos caixa branca, sendo assim diferente da nossa abordagem. No entanto, ao obter melhor desempenho com SVM, teve resultado que foi similar ao obtido na presente abordagem, na qual o SVM foi utilizado apenas para servir de *baseline* para os algoritmos Caixa-branca.

Uma abordagem para detecção de possíveis sintomas de baixo desempenho de alunos em e-learning foi proposta por AGAPITO et al. (2009). O método contém duas etapas principais: geração das regras de produção do algoritmo C4.5 e filtragem das regras mais representativas, o que poderia indicar baixo desempenho dos alunos. Além disso, a abordagem foi avaliada com os arquivos de log de atividades do estudante com duas versões de um sistema de questionário baseado na Web. Algumas regras apontaram que os alunos tinham dificuldades com atividades do curso, isso já se mostrando um informação relevante para o instrutor ou designer do curso, pois potencialmente eles podem utilize-las como insumos em suas decisões para melhorar o curso, por exemplo, adicionando novas atividades ou modificando as atividades existentes ou mesmo atuando na estrutura do curso.

WATSON et al. (2013) apresentam uma abordagem, chamada Watwin, para predição de desempenho de estudante em um curso de programação, considerando-se métricas sobre tempo de execução, linhas de código e mensagens de erro. Os resultados obtidos para indicar falhas deram uma acurácia de 75%, o que significa um resultado razoável. Esta abordagem certamente poderia ser incrementada com aumento de acurácia, incluindo-se outros atributos acadêmicos.

Uma pesquisa pouco relacionada, pois foca apenas em seleção de atributos, voltada para a predição do sucesso acadêmico, é a apresentada por ROY et al. (2018). Um diferencial desta pesquisa é a preocupação com a o direcionamento dos estudantes para a melhoria de suas capacidades e habilidades, para que possam alcançar a aprovação. Roy et al (2018) utiliza atributos socioeconômicos e acadêmicos, além de técnicas de classificação: SVM e Rede Neural. Nesta abordagem, portanto, não se investiu em algoritmo caixa-branca, não revelando preocupação com a explicabilidade do modelo.

Um outro trabalho que busca uma análise temporal, porém focado em um ambiente de aprendizagem do ensino à distância, o Moodle, foi realizado por BURGOS et al. (2018). Neste, o objetivo foi encontrar padrões com mineração de dados dos alunos de um curso à distância, permitindo a predição antecipada dos que irão abandonar o curso, dando assim um suporte ao professor do ensino à distância. A atenção aqui está direcionada para os resultados alcançados de desempenho das técnicas comparadas: observou-se que todas as técnicas alcançaram acurácia superior a 70% a partir da 8ª semana. Em princípio, trata-se ainda de um patamar de acurácia apenas razoável, numa antecedência temporal relevante.

Na pesquisa realizada por REDA et al. (2018), investe-se no desempenho acadêmico de estudantes, comparando-se os desempenhos das técnicas Naive Bayes, Árvore de Decisão e Multi-Layer Perceptron (MLP). A partir dos resultados encontrados por Reda, observa-se um destaque para a técnica de árvore de decisão, que alcançou um patamar destacável de 97,69 em acurácia.

O estudo realizado por YASSEIN et al. (2017) trata também da seleção de atributos com técnicas de mineração de dados educacionais, utilizando-se o pacote de software SPSS - Statistical Package for Social Sciences. Apesar da utilização de poucos atributos, mesmo assim, com um método de seleção de atributos conseguiu-se identificar aqueles mais relevantes. Também nesse mesmo percurso, inclusive utilizando o SPSS, em Bezerra et al (2016), abordou-se um problema de evasão de estudantes, buscando-se a extração de conhecimento a partir dos dados do censo escolar disponibilizado pelo INEP, visando identificar o perfil do aluno evasor e estimar a propensão à evasão através de Árvore de Decisão, Indução de Regras e Regressão Logística. Os

resultados mostraram que fatores como idade, turno das aulas e região geográfica das escolas influenciam fortemente a evasão. É interessante comparar tais fatores com os que mostram influência para a aprovação, explorados no experimento 1 deste trabalho.

Em relação às técnicas de Seleção de Atributos, observa-se uma descrição bastante focada e rica no trabalho de ZAFFAR (2017). Nele, foi feita uma análise de desempenho da seleção de atributos em dados de estudantes, mais precisamente, o ganho de desempenho alcançado a partir da remoção dos atributos menos relevantes. Esta metodologia também é abordada no presente trabalho, buscando-se o mesmo objetivo de comparação de ganho em desempenho.

Como visto anteriormente, existem muitas abordagens para estudar fenômeno de insucesso dos alunos a partir de técnicas de mineração de dados. Embora essas abordagens tenham apresentado formas promissoras para identificar os alunos que possam atingir o insucesso, elas são limitadas em termos de predição de insucesso atendendo os requisitos da presente abordagem: precisão alta, antecedência e automaticidade. Além disso, diferente desta proposta, boa parte destas abordagens não investem adequadamente na etapa de pré-processamento de dados, nem no ajuste fino nos algoritmos buscando mais eficácia das técnicas. Nos experimentos realizados, avaliou-se o desempenho das técnicas usadas no estudo para predição de desempenho acadêmico em cursos de programação, bem como se investiu em pré-processamento, inclusive constando sua influência positiva no resultado final, além de investir também em técnicas de ajuste fino nos algoritmos de predição, sempre conseguindo melhorias nos resultados finais.

MARTINHO et al.(2014) mostram os potenciais de um sistema inteligente desenvolvido para predição de grupos de estudantes em risco de evasão, usando uma rede Neural Fuzzy-Artmap. A base de dados para predição consistiu de fatores demográficos, fatores internos e externos à escola. Os dados demográficos tais como gênero, etnia, estado civil, renda familiar, turno de estudo, meio de transporte, etc., foram obtidos dos questionários socioeconômicos preenchidos pelos alunos na inscrição nos exames de seleção. Os dados acadêmicos foram obtidos do sistema de gerenciamento

acadêmico da instituição. A análise dos resultados mostrou acurácia global de 76%. Comparado à solução preditiva que vem se desenvolvendo na presente pesquisa de tese, percebe-se que este trabalho em avaliação tem um custo maior sobre coleta de dados, envolvendo até questionários, tendo ainda uma acurácia bem abaixo do que foi conseguido na proposta em apreço.

BAYER et al.(2012) utilizam um método para classificar estudantes em risco de insucesso no curso, considerando-se dados pessoais dos estudantes, acrescidos de dados relacionados a comportamentos sociais. Eles investem na fase de pré-processamento e avaliam a efetividade de sete algoritmos tentando selecionar o de melhor resultado. Nisto, conseguiu-se uma acurácia de 93,51%. No entanto, diferente da presente proposta que conseguiu boa acurácia atendendo ao requisito de antecedência, este trabalho alcançou essa boa acurácia somente no final do curso, o que torna tardia uma possível ação dos professores para evitar aspectos do insucesso.

Em (ER, 2012) propõe-se uma abordagem para predição de desempenho de estudantes, considerando-se três técnicas: k-NN, Árvore de Decisão e Naive Bayes. O experimento foi realizado em um curso de educação à distância e foi realizada em três etapas, que correspondem a diferentes estágios de um semestre. Em cada passo, novas instâncias foram adicionados às fontes de dados até que o curso alcançou a fase final. Os resultados revelaram que o modelo foi capaz de atingir uma eficácia de até 85%.

GOTTARDO et al. (2012) utilizaram algoritmos de classificação para identificar quais parâmetros de uma base de dados de interações de estudantes em um ambiente virtual possibilitariam uma inferência sobre o desempenho final dos alunos. Os resultados iniciais obtidos na pesquisa apontaram resultados satisfatórios, com acurácia dos algoritmos na ordem de 76%, com um conjunto amplo de atributos que representem de forma abrangente e generalizável um estudante, considerando a diversidade de cursos EAD existentes.

GURULER et al (2010) exploraram os fatores que têm impacto sobre o sucesso de estudantes universitários. Eles usaram uma ferramenta específica (MUSKUP DM) para fins de classificação. Os resultados revelaram que determinadas informações demográficas do estudante e o seu respectivo nível

de renda familiar estavam associados com o seu sucesso no curso.

Uma abordagem para detecção de possíveis sintomas de baixo desempenho de alunos e-learning foi proposta por AGAPITO et al.(2009). O método contém duas etapas principais: geração das regras de produção do algoritmo C4.5 e filtragem das regras mais representativas, o que poderia indicar baixo desempenho dos alunos. Além disso, a abordagem foi avaliada com os arquivos de log de atividades do estudante com duas versões de um sistema de questionário baseado na Web. Algumas regras apontaram que os alunos tinham dificuldades com atividades do curso. Essa informação pode ser relevante para o instrutor ou designer do curso, porque eles podem melhorar o curso adicionando novas atividades ou modificando as atividades existentes ou mesmo atuando na estrutura do curso.

O trabalho de LYKOURENTZOU et. al (2009) apresentou um método de previsão de abandono dos cursos em e-learning baseado em três técnicas de aprendizado de máquina mais comuns e nos dados detalhados de estudantes. As técnicas de aprendizado de máquina utilizadas foram redes neurais feedforward, máquinas de vetor de suporte (SVM) e um conjunto probabilístico simplificado fuzzy ARTMAP. Como uma única técnica pode falhar na classificação com precisão de alguns alunos de e-learning, enquanto outra pode ter sucesso, três esquemas de decisão, que combinaram de diferentes maneiras os resultados das três técnicas de aprendizado de máquina, também foram testados. As estimativas produzidas por cada técnica de aprendizado de máquina, bem como os produzidos por cada esquema decisão foram comparados em termos de precisão global, sensibilidade e precisão. Os resultados experimentais indicaram que a combinação dos resultados das três técnicas levou a uma identificação mais precisa e rápida de alunos evadidos.

MANHÃES et al. (2011) avaliaram o uso da EDM para previsão de estudantes com risco de evasão em uma universidade, por meio de três experimentos nos quais foram aplicados dez algoritmos de classificação sobre uma base de dados dos alunos de graduação num curso de Engenharia Civil. Os resultados mostraram que, utilizando as primeiras notas semestrais dos calouros, é possível identificar com acurácia média variando entre 75 a 80%, a situação final do aluno no curso.

Modelagem e previsão de desempenho do estudante - A modelagem orientada para representar e antecipar o desempenho do estudante é um dos alvos favoritos de abordagens de EDM. Vários indicadores de desempenho são possíveis de serem modelados, tais como: eficiência, avaliação, realização, competência, utilização de recursos, tempo decorrido, exatidão, deficiências, entre outros O objetivo é estimar o quanto bom o aluno é ou será capaz de realizar uma determinada tarefa, chegar a uma meta de aprendizagem específica ou dar resposta adequada a uma situação de aprendizagem particular (PEÑA-AYALA, 2014a).

SANTOS et al. (2012) relataram um estudo de caso sobre a aplicação de técnicas de mineração de dados (agrupamento e classificação) que permitem, em estágios anteriores às avaliações somativas, identificar alunos que têm maior risco de reprovação. Os dados que sustentam a abordagem proposta são oriundos de avaliações formativas aplicadas no decorrer da disciplina por meio do Moodle. Os resultados mostraram que os modelos criados permitem a identificação da propensão à reprovação com taxa de acerto em torno de 69%.

GOTTARDO et al.(2013) aplicaram técnicas de balanceamento de classes para melhorar os resultados de estimativas de desempenho futuro de estudantes, considerando cenários nos quais a quantidade de instâncias das classes é desbalanceado. Foi utilizada uma técnica conhecida como SMOTE (Synthetic Minority Over-sampling Technique) (CHAWLA et al., 2002), uma técnica que pode ser utilizada para ajustar a frequência relativa entre classes majoritárias e minoritárias nos dados. Os resultados obtidos apontam para a viabilidade da aplicação de técnica para identificar grupos de estudantes com maior risco de reprovação. por iniciantes como base para a geração de sugestão automática para tutores de programação. Foram usadas técnicas de EDM e aprendizado de máquina para automatizar a criação de sugestões e dicas a partir do agrupamento dos dados anteriores de estudantes na resolução de problemas semelhantes. Resultados preliminares mostraram que essa abordagem tem potencial para ser uma fonte para geração automática de dicas para programadores novatos.

3.1 Síntese

Verificou-se na discussão realizada neste capítulo trabalhos que são mais fortemente relacionados ao aqui proposto, considerando-se o todo da proposta, seguindo-se de alguns outros com similaridade em apenas alguns aspectos específicos da solução proposta, ou os que ainda possuem um foco diferente. Finalmente, ressalta-se que entre os trabalhos aqui discutidos, os mais fortemente relacionados ao aqui proposto, notadamente: MARQUEZ-VERA et al. (2016), MARQUEZ-VERA et al. (2013), foram selecionados e melhor discutidos nos dois capítulos seguintes,inclusive em comparação à nossa proposta.

4 ABORDAGEM PREDITIVA: ESTUDOS E RESULTADOS PRELIMINARES

Neste capítulo se inicia o percurso metodológico para alcançar os objetivos propostos, comprometidos com a busca por qualidade na informação obtida pelo uso de técnicas de mineração de dados para predizer o comportamento dos estudantes com respeito ao desempenho acadêmico, identificando-se tendências ao insucesso e ao sucesso. Assim, nesta etapa da abordagem desenvolvida, pretendeu-se primeiramente, como parte do percurso metodológico para alcançar parte dos objetivos propostos, realizar estudos exploratórios sobre a importância das técnicas de mineração de dados para a tarefa de predição, verificando-se a efetividade dos modelos analisados. Antes, porém, investiu-se técnicas de preparação de dados, incluindo-se, especialmente, técnicas de seleção de atributos, prosseguindo-se com balanceamento nas classes envolvidas. Entre outros, na intenção de obter informação oportuna e certa, três requisitos são buscados nesta abordagem preditiva: identificação com antecedência, de forma automática e com alta confiança. O encaminhamento dado na busca de uma tal solução de modelo preditivo realizou-se com um estudo sobre uso de técnicas de mineração de dados para a tarefa de predição, tentando responder primeiramente uma questão de viabilidade de uso destas técnicas para responder adequadamente ao problema posto. Além disso, objetivou-se comparar as principais métricas de avaliação aplicadas aos modelos preditivos, tanto caixa branca quanto caixa preta, e, a partir dessa análise, escolher qual modelo a ser adotado, nas demais fases da pesquisa, como a solução mais efetiva ao problema de predição em pauta. O detalhamento deste procedimento experimental é descrito a seguir.

4.1 Descrição da abordagem e Metodologia

Os quatro estudos experimentais descritos neste capítulo seguem uma abordagem, consistindo de um procedimento dotado de 4 etapas, conforme

ilustrado no diagrama da Figura 4.1, quais sejam:

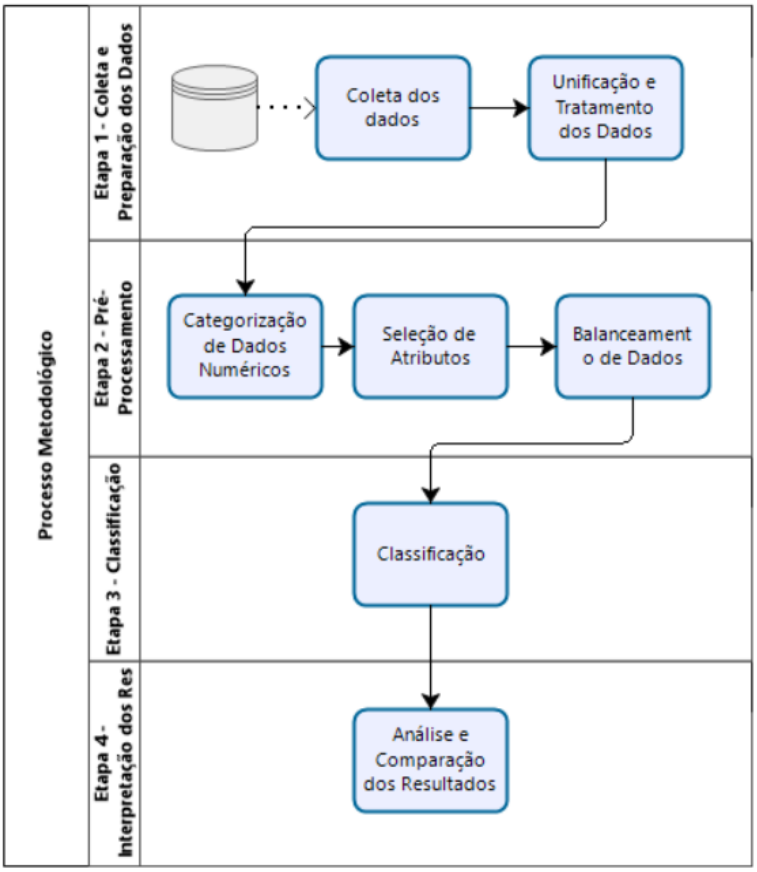
1. **Coleta e Preparação dos Dados**: Nesta etapa, faz-se o planejamento e coleta dos dados, sobre os estudantes, verificando-se aqueles que são necessários para abordar o problema em foco, bem como realiza-se a preparação e organização dos dados, considerando-se as diferentes fontes disponíveis. Os dados coletados são então integrados em uma base de dados única.

2. **Pré-Processamento:** Nesta etapa, os dados são preparados para se aplicar os algoritmos de predição. Assim, a partir da base de dados integrada, são necessárias alguns ajustes na base de dados, incluindo-se operações, tais como: limpeza de dados inconsistentes, transformação de variáveis, categorização. Em seguida, no intuito de melhorar o desempenho dos algoritmos de predição, são executados os algoritmos de Seleção de Atributos de forma a de manter apenas os atributos que possuem relevância significativa para a predição do sucesso e eliminar os demais, assim como aplica-se um balanceamento de dados (FACELI et al, 2011).

3. **Classificação:** É nesta etapa que os algoritmos de predição são executados. Adicionalmente, investe ainda em mecanismos de classificação sensível a custo, visando resolver problemas de desbalanceamento de dados.

4. **Interpretação dos Resultados**: Por fim, a partir dos resultados obtidos, faz-se uma análise comparativa entre os desempenhos dos algoritmos para determinar quem oferece os melhores resultados.

Figura 4.1 -Diagrama da Abordagem Metodológica Proposta



Fonte: Adaptado de Marquez-Vera et al. (2013)

Planejamento

Nos dois estudos a seguir, após investimento no entendimento e preparação dos dados, optou-se por explorar quatro técnicas de predição operando sobre dados de duas fontes distintas e independentes: uma oriunda de uma disciplina executada na modalidade à distância em um curso de graduação em sistema de informação e a outra de uma disciplina presencial de um curso de graduação em ciência da computação, ambos da Universidade Federal de Alagoas. Assim, foram analisadas as quatro seguintes categorias de classificadores: Indução de Árvore de Decisão, Naïve Bayes, Máquina de Vetores de Suporte e Redes Neurais. Essas escolhas foram baseadas primeiramente em uma tentativa de cobrir métodos diferentes, a exemplo métodos baseados em otimização (ex.: SVM e Redes Neurais), métodos baseados em busca (ex.: Árvore de Decisão), métodos probabilísticos (ex.:

Naive Bayes), métodos baseados em distância (ex.: KNN), isso tudo em sintonia com importantes trabalhos de revisão de literatura de EDM, que apontam esses classificadores como os predominantes em pesquisas na área.

Com base no propósito de investigar experimentalmente a efetividade dos modelos preditivos, comparando os seus desempenhos na tarefa de identificação, tão cedo quanto possível, de estudantes em risco de insucesso em disciplinas de programação introdutória, as seguintes questões de pesquisa foram elencadas para serem respondidas ao longo desta pesquisa, quais sejam:

Questão 1. Qual a eficácia dos algoritmos de predição baseados em técnicas de mineração de dados para identificar estudantes propensos ao insucesso?

A intenção com esta questão foi a de avaliar a eficácia dos algoritmos de predição que têm sido utilizados por abordagens existentes para identificar os estudantes propensos ao insucesso. Para responder a esta questão, os algoritmos de predição mencionados foram aplicados nos conjuntos de dados e, em seguida, a medida de f-measure foi utilizada para avaliar a eficácia de tais técnicas, além do que outras métricas de avaliação foram também usadas, conforme estão com os resultados exibidos mais adiante.

Questão 2. As etapas de pré-processamento de dados são capazes de aumentar em quanto a eficácia dos algoritmos de predição?

Esta questão teve por objetivo quantificar o quanto a eficácia destes algoritmos aumentava após a realização do pré-processamento de dados. Para responder esta questão, foi realizado o pré-processamnto nos conjuntos de dados utilizados neste trabalho, em seguida, foram aplicados os algoritmos de predição sobre os conjuntos de dados, avaliando a eficácia dessas técnicas e comparando esses resultados com a eficácia obtida executando as mesmas técnicas sobre o conjunto de dados, sem o pré-processamento.

Questão 3. A fase de ajustes finos nos algoritmos aumenta em quanto mais a eficácia dos algoritmos de predição?

Esta questão teve por objetivo analisar o quanto a eficácia dos algoritmos de predição poderia aumentar após a realização de ajustes finos em seus

parâmetros. Para responder a 3ª Questão, foram realizados os ajustes finos nos algoritmos utilizados e em seguida, foram executados os algoritmos de predição no conjunto de dados pré-processados. Após isso, avaliou-se a eficácia dos algoritmos e em seguida comparou-se suas eficácias relativamente aos resultados obtidos pela realização dos algoritmos de predição, sem o ajuste fino.

Questão 4. Depois de realizar o pré-processamento dos dados e os ajustes finos nos algoritmos de predição, quais das técnicas se mostraram mais eficazes na identificação antecipada dos estudantes propensos ao insucesso?

Esta 4ª Questão visou encontrar a técnica mais eficaz para identificação o mais cedo possível de estudantes susceptíveis ao insucesso. Para responder essa questão, foi analisada e comparada a eficácia das técnicas de algoritmos de predição após a realização das fases de pré-processamento de dados e de ajustes finos nos parâmetros dos algoritmos.

Questão 5. Quais os atributos mais relevantes para predição antecipada dos estudantes propensos ao insucesso?

A pesquisa para responder as cinco questões postas foi conduzida primeiramente em dois estudos, um sobre a disciplina na modalidade a distância e na modalidade presencial. Em cada estudo investiu-se em cinco grandes etapas: entendimento dos dados, preparação dos dados, modelagem preditiva e avaliação. No que se segue, há uma descrição de cada uma dessas duas fontes de dados e das características associadas a elas. Em seguida, prossegue-se com mais um estudo, desta vez ampliando o elenco dos classificadores, investindo-se em métodos baseados em árvore e em regras.

4.2 Predição de Desempenho: Estudo I

O propósito deste estudo avaliativo foi o de abordar as cinco questões de pesquisa mencionadas acima, investindo em técnicas de mineração de dados para predição de desempenho de estudantes do curso de graduação em Sistema de Informação da UFAL, na modalidade a distância, cursando a

disciplina Algoritmo e Estrutura de Dados I, sendo esta uma primeira disciplina sobre programação neste curso. Os dados usados neste estudo para classificar os estudantes foram oriundos de duas fontes de dados diferentes: Um ambiente virtual de aprendizagem, no caso o banco de dados do ambiente Moodle UFAL (2014) e mais o Sistema de Informação para o Ensino (SIE) (UFAL, 2014), que é o sistema de controle acadêmico usado pela universidade.

4.2.1 Método

Participantes

Ao todo, os participantes foram 262 estudantes, referentes ao ano 2013, tendo a disciplina em pauta uma duração de oito semanas, ocorrendo simultaneamente nos pólos das cidades de Maceió, Maragogi, Santana do Ipanema, Olho d'água das Flores e Arapiraca. Nesta disciplina, os estudantes eram avaliados semanalmente, recebendo notas de acordo com suas atividades desenvolvidas e mais duas avaliações aplicadas na quinta e na última semana do curso. A disciplina foi realizada com o uso do sistema Moodle.

Dados

Iniciando-se o processo de entendimento dos dados, envolvendo coleta e preparação dos dados, investiu-se na integração de dados provenientes dessas duas fontes, gerando uma única fonte, resultando nos seguintes atributos: Id, idade, gênero, estado civil, cidade onde realiza o curso, renda, matrícula, semestre, turma, período, ano de ingresso no curso dos estudantes, frequência de acesso ao AVA, participação nos fóruns de discussão, quantidade darquivos enviados e baixados do AVA, uso das ferramentas educacionais fornecidas pelo AVA, tais como: blog, glossário, quiz, wiki, message, notas relacionadas com as atividades desenvolvidas dentro do AVA, notas dos estudantes na disciplina (desempenho do estudante nas atividades semanais e testes) e a *status* do estudante na disciplina (aprovado ou reprovado).

Preparação dos dados

Após a fase de integração, prosseguiu-se com as etapas de limpeza de dados, transformação e seleção de atributos. Os atributos selecionados estão descritos na Tabela 2.

Tabela 2: Atributos Selecionados na modalidade de ensino a distância

Atributos	Descrição	Tipo de dado	Domínio
1ª Avaliação	Nota da primeira Avaliação	Numérico	[0,10]
5ª Semana	Nota da quinta semana	Numérico	[0,10]
2ª Semana	Nota da segunda semana	Numérico	[0,10]
4ª Semana	Nota da quarta semana	Numérico	[0,10]
3ª Semana	Nota da terceira semana	Numérico	[0,10]
Blog	Quantidade de postagem e visualização no blog	Numérico	[0,...]
1ª Semana	Nota da primeira semana	Numérico	[0,10]
Fórum	Quantidade de postagem e visualização no fórum	Numérico	[0,...]
Acessos	Quantidade de acessos ao AVA	Numérico	[0,...]
Assign	Quantidade de arquivos enviados e baixados	Numérico	[0,...]
Cidade	Cidade na qual o aluno reside	Caractere	[string]
Message	Quantidade de mensagens enviadas	Numérico	[0,...]
Wiki	Quantidade de mensagens enviadas	Numérico	[0,...]
Sexo	Gênero do estudante	Caractere	[M,F]
Estado	Estado Civil do estudante	Caractere	[S,C,V,D]
Idade	Idade do estudante	Numérico	[0,99]
Status	Status da disciplina (Aprovado/Reprovado)	Caractere	[aprovado,reprovado]

Fonte: **Adaptada de Costa et al. (2017)**

Construção e avaliação dos modelos preditivos

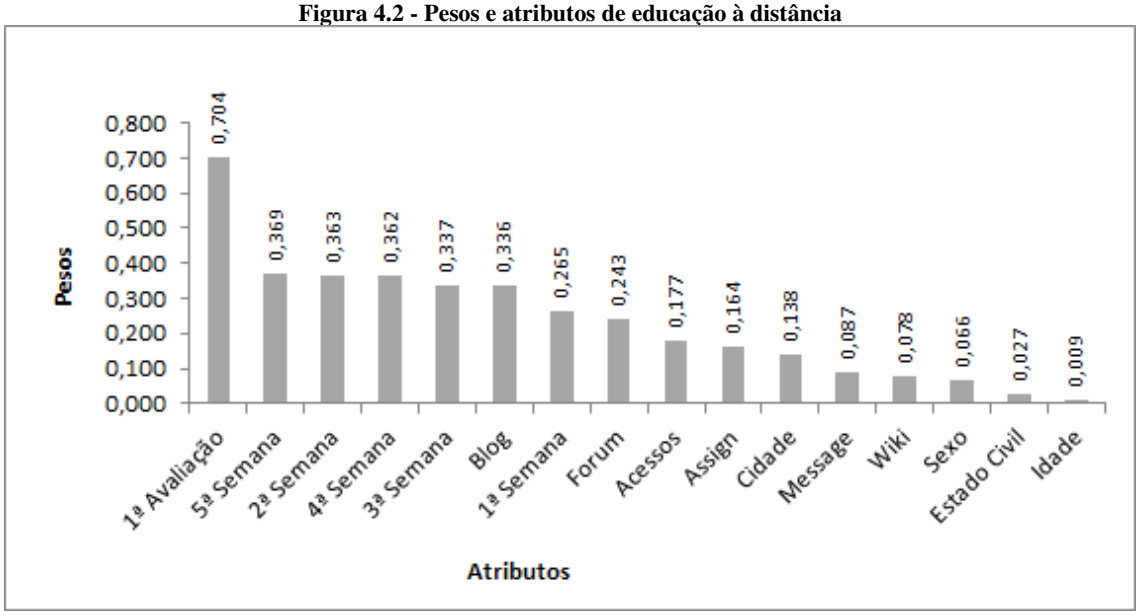
Nesta fase, construiu-se os modelos preditivos, avaliando-os e comparando-os, considerando-se além do KNN os seguintes classificadores: árvore de Decisão, SVM, Rede Neural e Nayve Bayes. O treinamento e teste foi realizado separadamente usando o método validação cruzada 10-fold. Ressalta-se ainda que se aplicou o processo de seleção de atributos e em seguida, investiu-se no balanceamento da distribuição de classe, pois a base de dados se mostrou desbalanceada, dado que a quantidade de estudantes reprovados era bem menor que a quantidade de aprovados. Para realização deste procedimento, foi utilizado o software WEKA, o qual disponibiliza diversos algoritmos de balanceamento dos dados, tendo sido escolhido o que se baseia na técnica SMOTE, a qual se mostrou mais apropriada para a realização desta tarefa. Aplicou-se ainda os mecanismos de ajustes finos nos algoritmos. Estas etapas estão descritas a seguir.

Seleção dos atributos

O método de seleção de atributos foi utilizado para escolher o melhor subconjunto de atributos, usando a técnica ganho de informação, conforme Capítulo 2, sendo aqui operacionalizada pela implementação InfoGain-AttributeEval, do pacote WEKA. Convém aqui lembrar que quando se realiza a seleção de atributos, espera-se que atributos irrelevantes e redundantes sejam removidos, o que normalmente leva um aumento da precisão do método de aprendizado. Além disso, alguns desses atributos não irão ser significativo para a classificação e é provável que alguns desses atributos sejam correlacionados. O software WEKA dispõe de vários algoritmos de seleção de atributos, que utilizam diferentes métodos, já discutidos no Capítulo 2. Durante a realização do trabalho foram testados vários algoritmos, tais como: CfsSubsetEval, ChiSquaredAttributeEval, Consistency-SubsetEval, FilteredAttributeEval, OneRAt- tributeEval, FilteredSubsetEval, GainRatioAttributeEval, InfoGainAttributeEval, ReliefFAttributeEval,

SymmetricalUncert-AttributeEval (WITTEN, FRANK, HALL, 2011). No entanto, o que apresentou o melhor resultado em termos de precisão foi o algoritmo InfoGainAttributeEval (QUINLAN,1986), sendo então adotado e usado, produzindo resultado conforme ilustrado na Figura 4.2, ressaltando os pesos atribuidos a cada atributo selecionado.

Figura 4.2 - Pesos e atributos de educação à distância



Fonte: Adaptada de Costa et al. (2017)

Ajustes finos nos algoritmos

Estudos (Gunawan et al., 2011; Hutter et al., 2009) indicam que a eficácia de algumas técnicas de EDM pode ser aprimorada por meio do ajuste fino. A fim de corroborar esses estudos, realizou-se o ajuste fino das técnicas de EDM aqui utilizadas. Deste modo, usou-se as fontes de dados pré-processadas para comparar a eficácia das técnicas nas duas circunstâncias: sem o ajuste fino e, em seguida, com o ajuste fino.

Os desempenhos dos algoritmos de classificação são sensíveis aos ajustes de parâmetros, principalmente em problemas do mundo real (VIANA et al., 2007). Os métodos de escolha destes parâmetros variam amplamente e são conhecidos como *tunning*. Com o objetivo de aumentar a eficácia dos algoritmos, optou-se aqui por ajustar os parâmetros dos algoritmos utilizados neste trabalho.

(Naive Bayes) - O ajuste no algoritmo Naive Bayes foi realizada seguindo a abordagem descrita em (JOHN; LANGLEY, 1995), que usa um método baseado na estimativa do kernel para realizar o ajuste fino.

(Árvore de decisão) - De acordo com (WITTEN; FRANK; HALL, 2011), a eficácia do algoritmo J48 pode ser melhorada através da realização de um ajuste fino de dois parâmetros: (i) a quantidade de nós de folha; e (ii) a poda de árvore de decisão. Assim, no presente estudo, realizou-se alguns experimentos comparativos, a fim de encontrar os melhores valores para esses parâmetros.

(Rede Neural) – Realizou-se os ajustes de três parâmetros do algoritmo Rede Neural: (i) a taxa de aprendizagem (*learning rate*); (ii) o impulso aplicado aos pesos durante a sua atualização (*momentum*); (iii) o número de camadas escondidas existentes na rede (*hidden layers*). De acordo com (WITTEN; FRANK; HALL, 2011), o ajuste fino destes parâmetros pode melhorar a eficácia do algoritmo de rede neural.

(Máquina de Vetores de Suporte) - A primeira mudança nos parâmetros do algoritmo SVM foi em relação ao Kernel, sendo testadas 3 opções: polinomial, RBF e Sigmoide, sendo que quem apresentou o melhor resultado foi o Kernel gaussiano (RBF). Utilizou-se o método conhecido como Grid-Search[2] para buscar pelos melhores parâmetros através da análise dos resultados obtidos com a execução do algoritmo para um intervalo de parâmetros, uma vez que não se sabe previamente qual ou quais os melhores parâmetros para o problema em questão. Após o uso do método Grid-Search, ajustou-se os parâmetros manualmente com valores aproximados ao informado pelo método. Assim, o método Grid-Search serviu como heurística para os ajustes nos parâmetros do algoritmo SVM.

4.2.2 Desenvolvimento e resultados

Nesta seção, descreve-se de que forma os algoritmos foram executados, na tarefa de conseguir identificar os estudantes propensos ao insucesso. Os

[2] Esse método consiste em fazer uma busca exaustiva, combinando valores de determinados intervalos para cada parâmetro que se deseja otimizar . Para cada combinacão de parâmetros, o SVM deve ser executado. Ao fim da grid search, a combinacão de parâmetros que gerar melhor resultado é escolhida.

resultados estão mostrados e discutidos logo a seguir, tendo na Figura 4.2 uma visão geral de um comparativo, via uso da f-measure, dos quatro algoritmos utilizados, sem e com a realização do pré-processamento, incluindo a seleção de atributos, além da etapa com ajuste fino dos algoritmos. Mais detalhes desses resultados podem ser encontrados em Costa et al. (2017).

Levando-se em consideração que o objetivo é prever situação final do estudante com a maior antecedência possível, dentro da disciplina dada, executou-se os algoritmos separadamente sobre cada modalidade de ensino enriquecendo os dados semanalmente. Ou seja, os algoritmos foram executados em etapas; na primeira etapa foram incluídos dados somente até o último dia da primeira semana de aula, na segunda etapa, foram acrescentados dados até o último dia da segunda semana de aula, e assim consequentemente até chegar à semana da primeira avaliação.

Além do enriquecimento dos dados, os algoritmos foram executados em três fases distintas:

- Na primeira fase, todos os algoritmos foram executados sobre os conjuntos de dados com somente a integração de dados realizada sem a utilização de qualquer outra etapa de pré-processamento dos dados, ou sem **seleção automática de atributos;**
- Na segunda fase, os algoritmos foram executados após as etapas de limpeza dos dados, transformação dos dados**, seleção dos atributos e balanceamento de dados**;
- Finalmente, na última etapa os algoritmos foram executados após a realização dos **ajustes finos**.

Para realizar a avaliação da capacidade de generalização do modelo, todas as etapas foram avaliadas utilizando o método de validação cruzada k-fold, com k=10. Para melhor caracterizar a eficácia dos algoritmos analisados neste trabalho, decidimos adotar a métrica F-Measure.

Para realização deste experimento, tal como já informado, foi utilizada a ferramenta WEKA Experiment Environment (WEE)(WEKA, 2014). O WEE oferece três opções de estratificação da base de dados: i) Cross validation

(default), ii) Train/Test Percentage Split (data randomized) e iii) Train/Test Percentage Split (order preserved). Para obter significância estatística nos desempenhos dos algoritmos, o ambiente foi configurado com um número padrão de execuções. Por exemplo, cada algoritmo é executado 10 vezes e seu desempenho final é obtido a partir da média das execuções. No caso do método *10-fold cross-validation* significa que um classificador foi executado 100 vezes para os conjuntos de treinamento e teste.

Discussão dos Resultados

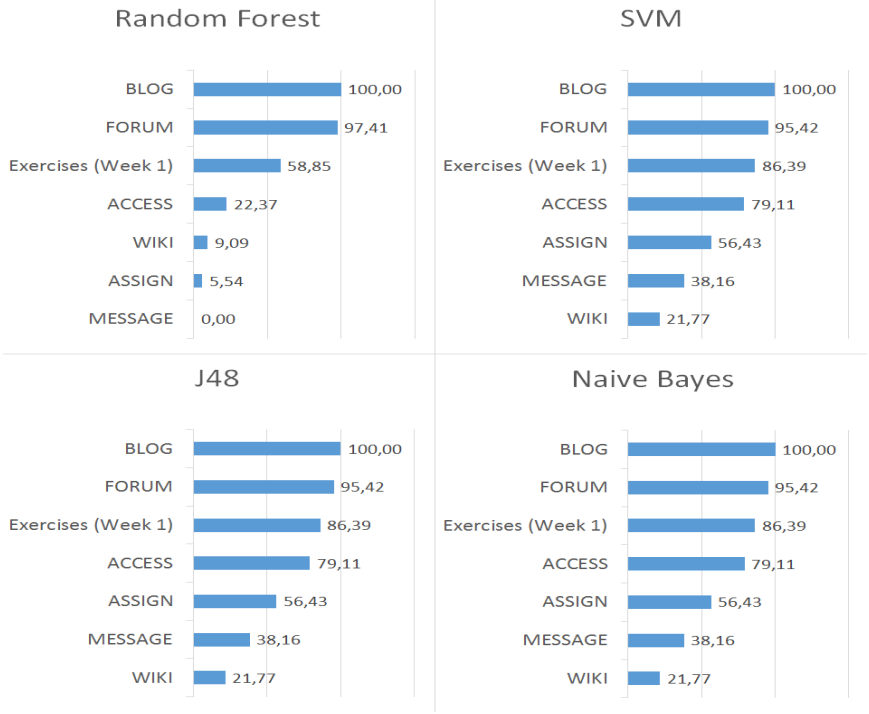
A seguir, apresenta-se uma discussão sobre os principais resultados do experimento, obtidos em cada semana. Primeiramente, mostrou-se a influência dos atributos descritos, sobre a predição de falhas de estudantes, considerando-se da primeira à quinta semana do curso quando o primeiro exame foi aplicado. Figuras 4.3 a 4.7 descrevem a importância de cada atributo sobre a predição de falhas dos estudantes usando as técnicas de mineração de dados em questão.

Primeira Semana

Na Figura 4.3, descreve-se a influência dos atributos sobre a predição de insucesso dos estudantes, usando as técnicas: Random Forest, SVM, Árvore de Decisão via J48 e Naive Bayes.

Observou-se que Blog, Forum, Exercises (Semana 1) e acesso tinham a mais alta influência sobre a predição realizada com todas as técnicas analisadas. Em particular, Blog alcançou uma influência de 100% em todas as análises quando se considerou a primeira semana do curso. Por outro lado, Wiki apresentou a mais baixa influência em 3 (SVM, J48 e Naive Bayes) das 4 técnicas analisadas. Assign alcançou a mais baixa influência no caso da técnica Random Forest.

Figura 4.3- Primeira semana



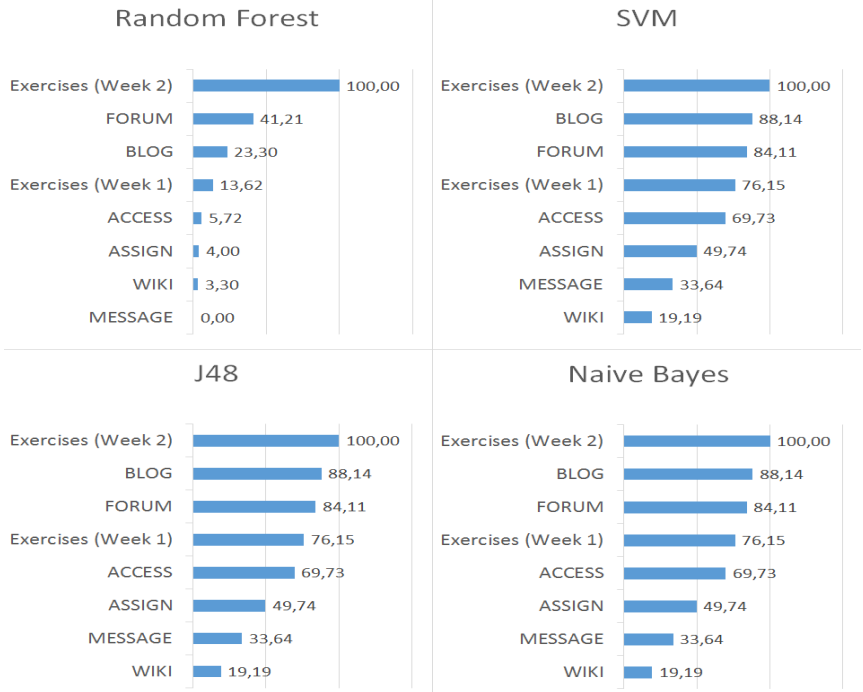
Fonte: Adaptada de Costa et al. (2017)

Segunda semana

Com relação à segunda semana, os atributos Exercícios (Semana 2), Blog, Forum, Exercícios (Semana 1) e Acesso tiveram a mais alta influência sobre a predição realizada pelas técnicas, tal como mostrado na Figura 4.4. Os exercícios (Semana 2) alcançaram uma influência igual a 100% em todos os casos.

Analogamente à primeira semana, o atributo Wiki alcançou uma menor influência preditiva. De fato, teve a pior influência em todos os casos analisados.

Figura 4.4- Segunda Semana



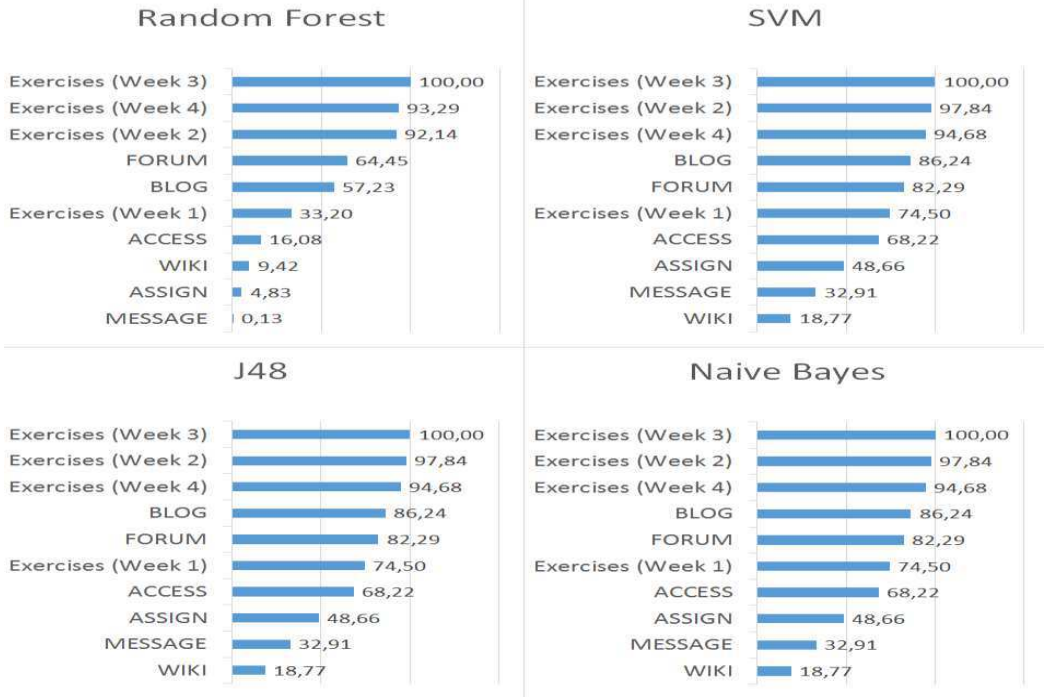
Fonte: Adaptada de Costa et al. (2017)

Terceira semana

De acordo com os resultados exibidos na Figura 4.5, observa-se que os Exercícios (Semana 3) tiveram uma influência de 100% na previsão realizada na terceira semana do curso. Note também que a influência alcançada pelos Exercícios (Semana 2) foi um pouco menor que os Exercícios (Semana 3).

Analogamente à análise referente às semanas anteriores, o Fórum, Blog, Exercícios (Semana 1) e Acesso também alcançaram uma alta influência na previsão. Novamente, a Wiki apresentou a pior influência em todos os casos analisados.

Figura 4.5- Terceira Semana



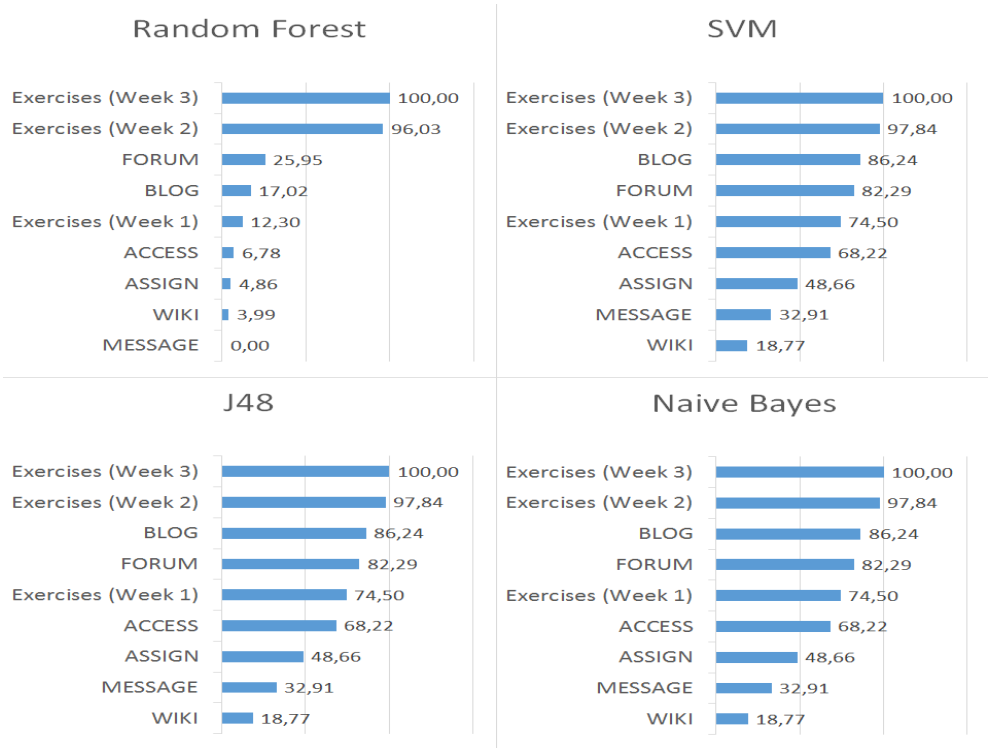
Fonte: Adaptada de Costa et al. (2017)

Quarta Semana

Assim como na Terceira Semana, o atributo Exercícios (Semana 4) teve a maior influência na Quarta Semana, atingindo 100% de influência em todos os casos, conforme ilustrado na Figura 4.6. Os valores de influência alcançados pelos Exercícios (Semana 2 e 4) os atributos foram ligeiramente inferiores aos Exercícios (Semana 3).

Mais uma vez, o Blog, Fórum, Exercícios (Semana 1) e Acesso tiveram uma grande influência nas previsões. Diferente da Segunda e Terceira Semana, o atributo Wiki teve a menor influência apenas no caso do SVM, J48 e Naive Bayes. O atributo Assign atingiu a pior influência no caso Random Forest.

Figura 4.6- Quarta Semana



Fonte: Adaptada de Costa et al. (2017)

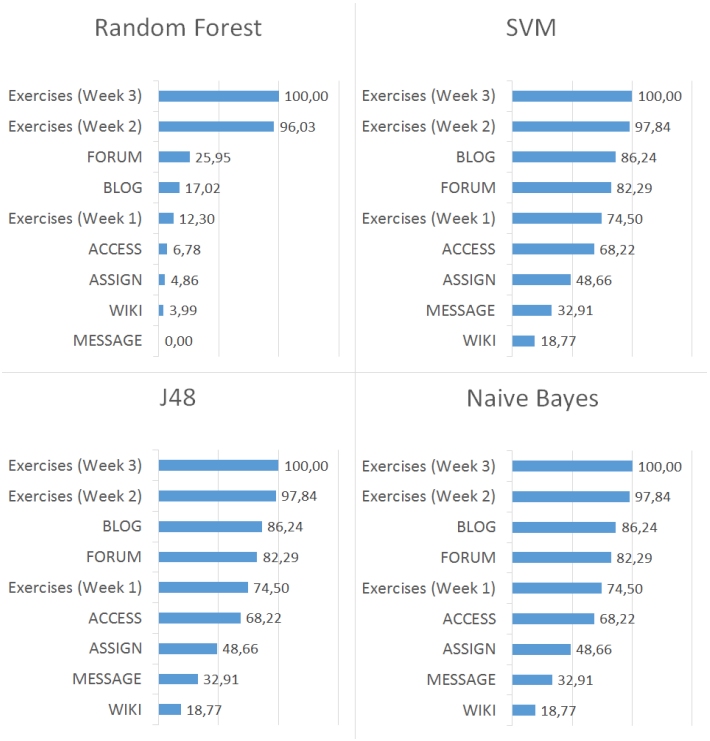
Quinta Semana

Na Quinta Semana, a grande maioria dos atributos investigados teve baixa influência nas predições pelo uso da Floresta Aleatória (Random Forest), conforme indicado na Figura 4.7. Note que os Exercícios (Semana 5) foram os principais responsáveis por influenciar a predição pelo uso desta técnica.

Por outro lado, nas demais técnicas (SVM, J48 e Naive Bayes), os atributos apresentaram comportamento semelhante ao observado nas semanas anteriores. Os exercícios realizados pelos alunos durante a semana atual tiveram maior influência na previsão. Note-se que os Exercícios (Semana 5) atingiram uma influência de 100% em todos os casos analisados.

Os exercícios realizados pelos alunos na segunda semana (Exercícios - Semana 2), terceira (Exercícios - Semana 3) e quarta (Exercícios - Semana 4) tiveram uma importância um pouco menor que os Exercícios (Semana 5). Da mesma forma que nas semanas anteriores, o Blog, o Fórum e o Access também alcançaram uma grande influência, e o Wiki teve a menor influência.

Figura 4.7 -Quinta semana



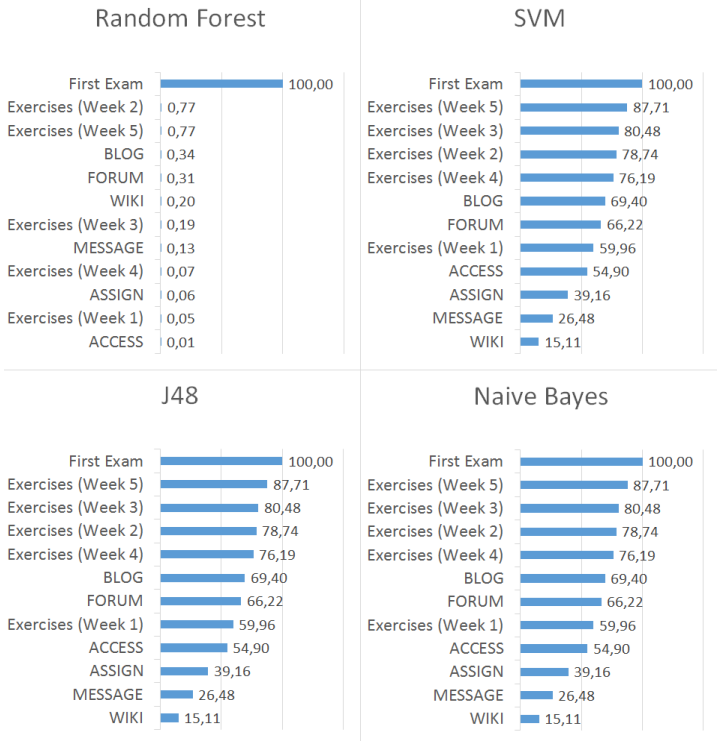
Fonte: Adaptada de Costa et al. (2017)

Primeira Avaliação

Após a aplicação do primeiro exame, observa-se na figura 4.8 que os resultados obtidos pelos alunos no exame passaram a ser os principais responsáveis por influenciar as previsões. De fato, o atributo Primeiro Exame alcançou uma influência de 100% em todos os casos analisados. Particularmente, na Floresta Aleatória, as predições foram totalmente determinadas por este atributo.

Em relação aos demais atributos, apresentaram comportamento semelhante às previsões realizadas nas semanas anteriores. Os exercícios realizados pelos alunos, assim como o Blog, Fórum e Acesso, tiveram grande influência nas previsões.

Figura 4.8- Primeira Avaliação



Fonte: Adaptada de Costa et al. (2017)

Resultados Gerais

Quanto à Questão 1, verificando a efetividade dos algoritmos na predição antecipada, conforme ilustrado na Figura 4.2, constatou-se que tais algoritmos apresentaram um desempenho que variou de 0,55 a 0,82, isso indicando que após a primeira semana do curso, conseguiu-se identificar os estudantes com tendência ao insucesso, com pelo menos 0,50. Ressalta-se ainda que os algoritmos de árvore de decisão exibiram melhor efetividade, alcançando uma medida f-measure igual a 0,82 no momento da primeira avaliação, ou seja, esse resultado foi alcançado a um momento de aproximadamente 50% do curso. Destaca-se que esses resultados enfatizados se referem ao investimento sem ainda ter empregado as técnicas de seleção de atributos, balanceamento e ajuste fino, pois tais aspectos estão relacionados às questões 2, 3 e 4, tendo, tal como já era esperado, desempenhos superiores.

Quanto às Questões 2 e 3, os resultados já foram exibidos e discutidos antriormente, revelando o que já era esperado em termos de incrementos nos desempenhos dos algoritmos, ressaltando-se apenas que a técnica Naïve

Bayes não apresentou melhoria significativa.

Quanto à Questão 4, observa-se, conforme ilustrado nas Figuras de 4.3 à 4.8, que os atributos relacionados aos exercícios e aos exames realizados pelos alunos, durante cada semana, tiveram maior influência na grande maioria dos casos na predição de insucesso dos alunos. Excetuando-se a Primeira Semana, o atributo Blog atingiu a maior influência. Nota-se também que o Blog, Fórum e Access tiveram uma forte influência em todos os casos analisados. Por outro lado, o atributo Wiki alcançou a menor influência na maioria dos casos. De fato, somente em 2 (como pode ser visto no caso do J48 na Primeira e Quarta Semanas) dos 24 casos analisados, o Wiki não alcançou o menor nível de influência.

Quanto à Questão 4, relativamente à influência do mecanismo de ajuste, verificou-se que, após a seleção de atributos e a realização do ajuste final, houve melhoria nos resultados, notadamente no SVM, com o qual se obteve 0,92 na métrica f-measure no momento da primeira avaliação, conforme mostrado na Figura 4.8, o que corresponde a um momento de realização de 50% do curso.

Quanto à Questão 5, nas Figuras de 4.3 à 4.7 já se ilustra um ranqueamento da relevância dos atributos relativamente à tarefa de predição.

Observa-se que os atributos relacionados aos exercícios e aos exames realizados pelos alunos durante a semana atual tiveram maior influência na grande maioria dos casos na predição de falhas dos alunos. Exceto na Primeira Semana, o atributo Blog atingiu a maior influência.

Note também que o Blog, Fórum e Access tiveram uma grande influência em todos os casos analisados. Por outro lado, o atributo Wiki alcançou a menor influência na grande maioria dos casos. De fato, somente em 2 (veja o caso do J48 na Primeira e Quarta Semanas) dos 24 casos analisados, o Wiki não alcançou o menor nível de influência.

Ameaças

Embora o experimento tenha fornecido evidências significativas sobre a

eficácia dos métodos de seleção com algoritmos preditivos para identificar o mais cedo possível os alunos que tenderão ao insucesso, ressaltam-se aqui algumas ameaças.

Em primeiro lugar, a fonte de dados utilizada no experimento em apreço representa dados sobre alunos de apenas um curso, em uma modalidade de educação a distância, em uma universidade pública brasileira. Portanto, os resultados da experiência não são gerais.

Em termos de métrica de avaliação dos modelos preditivos, adotou-se apenas a *f-measure* para caracterizar suas eficácias, mesmo tendo sido verificadas outras. Embora esta medida tenha sido amplamente utilizada em várias pesquisas observadas na literatura, outras medidas, tais como acurácia e Kappa, poderiam ser usadas.

4.3 Predição de Desempenho: Estudo II

Este estudo avaliativo tem o mesmo propósito do anterior, investindo em dados de estudantes do curso de ciência da computação da UFAL, em Maceió, na modalidade presencial, no ano 2014, cursando a disciplina Programação I, sendo esta uma primeira disciplina sobre programação. Assim, os dados usados neste estudo para classificar os estudantes foram oriundos de duas fontes de dados diferentes: Um ambiente de aprendizagem online, no caso o banco de dados do ambiente THE HUXLEY (2014) e o banco de dados do SIE (UFAL, 2014), que é o sistema de controle acadêmico usado pela universidade.

4.3.1 Método

Participantes

Os participantes que fizeram parte deste estudo totalizaram um banco de dados com registros de 161 estudantes de graduação que cursaram a disciplina de programação introdutória, em 2014, durante dezesseis semanas. Neste curso os estudantes foram avaliados semanalmente de acordo com as

atividades propostas, além de testes aplicados na quarta, oitava, decima segunda e décima sexta semanas do curso.

Dados

Iniciando-se o processo de entendimento dos dados, envolvendo coleta e preparação dos dados, investiu-se na integração de dados provenientes dessas duas fontes, gerando uma única fonte, resultando nos seguintes atributos: Id, idade, gênero, estado civil, cidade, renda, matrícula, período, disciplina, semestre, campus, ano de entrada no curso, status, quantidade de exercícios realizados, número de exercícios corretos, desempenho do estudante nas atividades semanais e testes.

Preparação dos Dados

Após a fase de integração prosseguiu-se com as etapas de limpeza de dados, transformação e seleção de atributos. Os atributos selecionados estão descritos na Tabela 3.

Tabela 3: Atributos Selecionados na modalidade de ensino presencial.

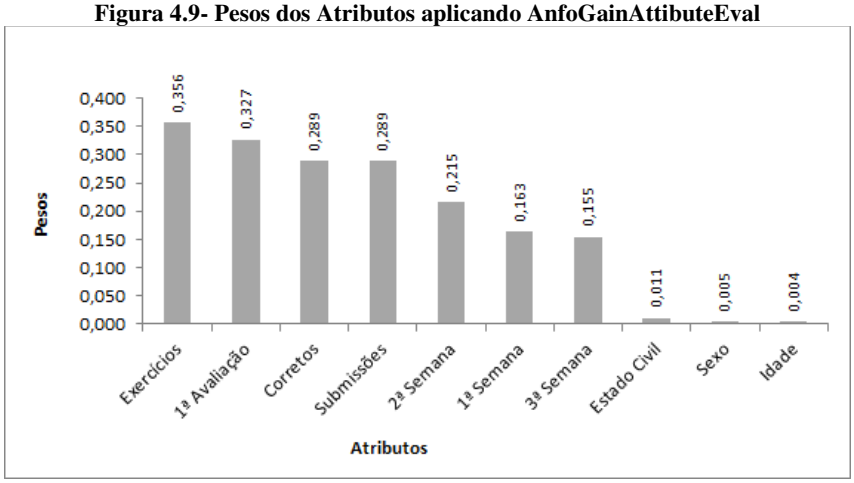
Atributos	Descrição	Tipo de Dado	Domínio
Problemas	Quantidade de Exercícios	Numérico	[0,..]
1ª Avaliação	Nota da primeira Avaliação	Numérico	[0,10]
Corretos	Total de Exercícios corretos	Numérico	[0,..]
Submissões	Total de submissões	Numérico	[0,..]
2ª Semana	Nota da segunda semana	Numérico	[0,10]
1ª Semana	Nota da primeira semana	Numérico	[0,10]
3ª Semana	Nota da terceira semana	Numérico	[0,10]
Estado Civil	Estado Civil	Caractere	String
Sexo	Sexo	Caractere	[C,S,D,V]
Idade	Idade	Numérico	[0,99]
Status	Status (Aprovado/Reprovado)	Caractere	[Aprovado, Reprovado]

Fonte: Elaborada pela autora

Seleção de atributos

Na Figura 4.9, exibe-se os pesos dos atributos obtidos mediante aplicação do algoritmo InfoGainAttributeEval.

Figura 4.9- Pesos dos Atributos aplicando AnfoGainAttibuteEval



Fonte: Adaptada de Costa et al. (2017)

4.3.2 Avaliação dos modelos e resultados do estudo 2

Na Figura 4.10 estão exibidos os resultados gerais do estudo 2, os quais se encontram discutidos em seguida.

Figura 4.10- Resultados Gerais- Ensino Presencial

Ensino presencial

Antes do Pré-processamento

	1ª Semana				2ªSemana				3ª Semana				1ª Avaliação – Sem Ajustes			
Classificador	NB	DT	RN	SVM	NB	DT	RN	SVM	NB	DT	RN	SVM	NB	DT	RN	SVM
Acurácia	57.17	74.5	68.33	71	59.67	73	70.17	65.5	51	73.17	71	77	70.33	73.5	70.5	79.83
Kappa	0.29	0.41	0.32	0.34	0.23	0.39	0.36	0.15	0.19	0.39	0.38	0.42	0.38	0.41	0.39	0.55
Precision	0.5	0.75	0.73	0.62	0.53	0.76	0.78	0.6	0.59	0.76	0.77	0.76	0.76	0.73	0.82	0.81
Recall	0.55	0.7	0.76	0.49	0.58	0.37	0.75	0.53	0.54	0.73	0.74 v	0.71	0.65	0.74	0.64	0.71
F-measure	0.5	0.7	0.72	0.52	0.53	0.79	0.74	0.56	0.54	0.78	0.73	0.73	0.67	0.73	0.68	0.75
ROC Area	0.59	0.65	0.55	0.57	0.72	0.71	0.75	0.58	0.73	0.72	0.76	0.78	0.81	0.75	0.81	0.79
PRC Area	0.71	0.69	0.75	0.61	0.84	0.75	0.86	0.67	0.84	0.76	0.87	0.86	0.88	0.79	0.89	0.82

Depois do Pré-processamento

	1ª Semana				2ªSemana				3ª Semana				1ª Avaliação			
Classificador	NB	DT	RN	SVM	NB	DT	RN	SVM	NB	DT	RN	SVM	NB	DT	RN	SVM
Acurácia	57.42	76.75	78.00	72.56	60.51	84.04	76.15	74.72	61.58	79.56	79.00	82.18	76.76	81.15	78.74	83.67
Kappa	0.14	0.52	0.55	0.39	0.20	0.67	0.51	0.44	0.22	0.57	0.57	0.59	0.52	0.61	0.55	0.65
Precision	0.51	0.78	0.79	0.82	0.55	0.88	0.79	0.85	0.7	0.82	0.79	0.93	0.78	0.83	0.82	0.88
Recall	0.56	0.72	0.72	0.42	0.58	0.78	0.75	0.48	0.74	0.72	0.75	0.62	0.68	0.74	0.67	0.73
F-measure	0.51	0.72	0.72	0.53	0.55	0.80	0.75	0.58	0.56	0.73	0.74	0.74	0.69	0.75	0.71	0.78
ROC Area	0.72	0.87	0.87	0.69	0.77	0.90	0.85	0.71	0.78	0.88	0.86	0.86	0.88	0.81	0.86	0.82
PRC Area	0.72	0.82	0.87	0.63	0.79	0.85	0.86	0.66	0.79	0.82	0.86	0.87	0.88	0.73	0.87	0.77

Depois dos ajustes Finos

	1ª Semana				2ªSemana				3ª Semana				1ª Avaliação			
Classificador	NB	DT	RN	SVM	NB	DT	RN	SVM	NB	DT	RN	SVM	NB	DT	RN	SVM
Acurácia	72.85	79.82	78.24	72,68	79.17	86.51	76.76	77,49	78.54	81.57	79.13	84,69	84,17	83.19	82.82	88,04
Kappa	0.48	0.59	0.55	0.39	0.58	0.72	0.52	0.51	0.55	0.61	0.57	0.64	0.60	0.65	0.64	0.75
Precision	0.67	0.79	0.78	0,83	0.77	0.91	0.77	0,88	0.32	0.85	0.80	0,96	0.89	0.87	0.88	0,94
Recall	0.89	0.80	0.74	0,42	0.81	0.78	0.69	0,56	0.59	0.72	0.74	0,62	0.74	0.75	0.70	0,78
F-measure	0.74	0.77	0.73	0,53	0.76	0.32	0.69	0,66	0.71	0.75	0.73	0,75	0.78	0.73	0.76	0,83
ROC Area	0.88	0.88	0.38	0.74	0.91	0.91	0.87	0.81	0.92	0.89	0.88	0.89	0.81	0.85	0.84	0.86
PRC Area	0.87	0.83	0.38	0.76	0.90	0.37	0.87	0.83	0.91	0.83	0.88	0.85	0.78	0.82	0.87	0.89

Fonte: Adaptada de Costa et al. (2017)

Resultados

A seguir, estão descritos os resultados obtidos para cada uma das questões de pesquisa em pauta.

Questão de Pesquisa 1. Qual a eficácia dos algoritmos de predição baseados em técnicas de mineração de dados para identificar estudantes propensos ao insucesso?

Para responder a esta questão foram executadas as quatro técnicas de classificação. Na Figura 4.11 exibe-se a eficácia (representada pela métrica f-measure) dos algoritmos de predição para identificar os alunos com tendência ao insucesso, considerando-se apenas os dados dos alunos até a primeira aplicação dos exames.

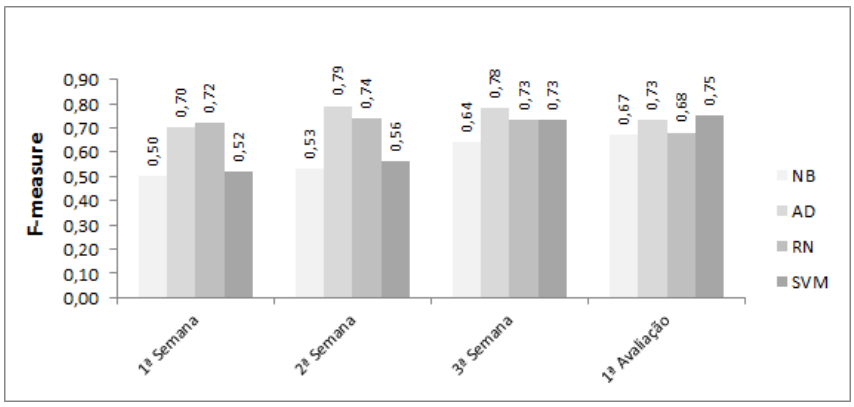
Observa-se que os algoritmos apresentaram uma efetividade que varia de

0,50 a 0,79. Esses resultados indicam que, após a primeira semana dos cursos, os algoritmos foram capazes de identificar com pelo menos 50% de eficácia, os alunos que provavelmente teriam insucesso.

Observa-se também que os algoritmos de Árvore de Decisão apresentaram uma melhor eficácia, atingindo um valor de f-measure igual a 0,82 após a aplicação do primeiro exame e 0,79 na segunda semana.

Dado que os cursos presenciais têm duração de 16 semanas, pode-se afirmar que a técnica Árvore de Decisão foi capaz de atingir uma eficácia igual a 79%, quando os alunos se encontravam em um momento equivalente a 25% da realização do curso. Portanto, os resultados apresentaram evidências de que as técnicas de predição analisadas nesses experimentos foram eficazes para identificar antecipadamente os alunos com tendência ao insucesso. No entanto, não se pode aqui assegurar a generalização de tais resultados para outros contextos de aplicação.

Figura 4.11- Eficácia dos métodos (NB - Naive Bayes; AD – Árvore de Decisão; RN – Rede Neural; SVM – Máquina de Vetor de Suporte)



Fonte: Adaptada de Costa et al. (2017).

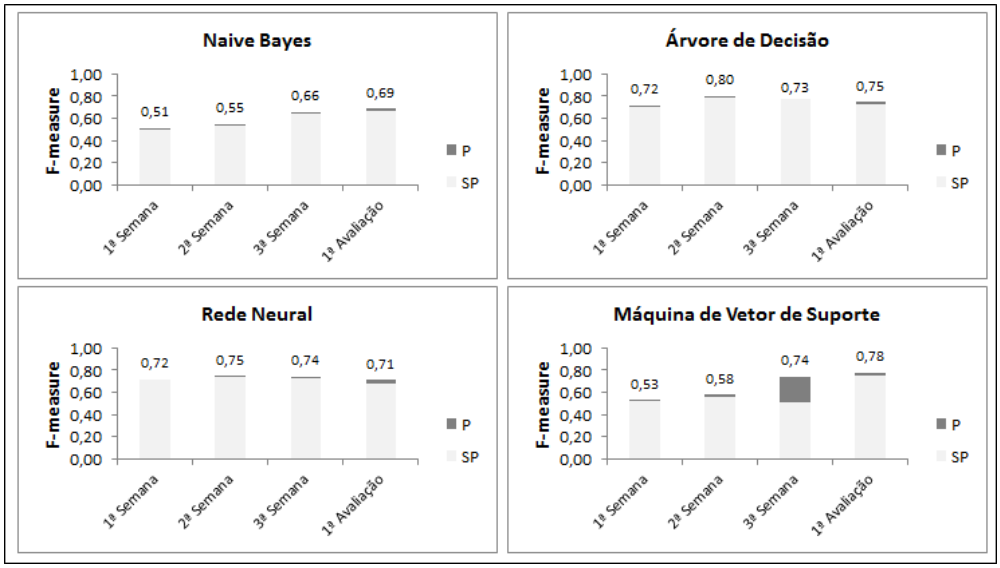
A seguir, mostra-se em quanto a eficácia dessas técnicas pode ser melhorada executando as etapas: pré-processamento de dados e ajuste fino de algoritmos.

Questão de Pesquisa 2. As etapas de pré-processamento de dados são capazes de aumentar em quanto a eficácia dos algoritmos de predição?

A fim de responder tal pergunta, realizou-se um pré-processamento sobre as duas fontes de dados usadas neste experimento, depois aplicou-se as

quatro técnicas de predição nas fontes de dados pré-processadas, avaliou-se então a eficácia dessas técnicas nas fontes de dados pré-processadas e, por fim, comparou-se esses resultados com os obtidos sem pré-processamento. Na Figura 4.12, apresenta-se os resultados comparativos da eficácia das quatro técnicas de predição aplicadas ao conjunto de dados

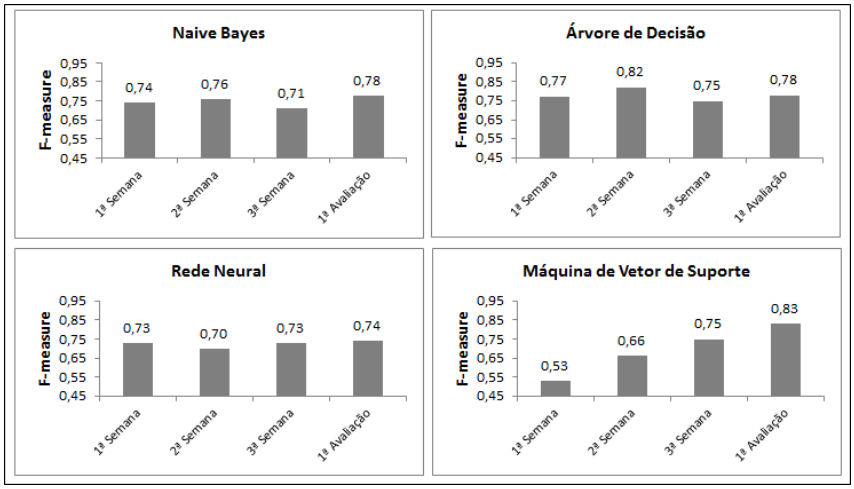
Figura 4.12- Resultados Comparativos da Eficácia dos Métodos sobre os dados sem e com pré-processamento.



Fonte: Adaptada de Costa et al. (2017).

Sobre a questão de pesquisa 3, os resultados obtidos com a aplicação dos mecanismos de ajustes finos estão exibidos na Figura 4.13, observando-se novamente melhorias no desempenho dos algoritmos, relativamente a esta questão de pesquisa.

Figura 4.13- Resultados comparativos da efetividade dos métodos EDM depois dos ajustes finos.



Fonte: Adaptada de Costa et al. (2017).

Questão de Pesquisa 4. Depois de realizar o pré-processamento dos dados e os ajustes finos nos algoritmos de predição, quais das técnicas se mostraram mais eficazes na identificação dos estudantes propensos ao insucesso?

Esta 4ª Questão visou encontrar a técnica mais eficaz para identificação o mais cedo possível de estudantes susceptíveis ao insucesso. Para responder esta questão, foi analisada e comparada a eficácia dos algoritmos de predição, após a realização das fases de pré-processamento de dados e de ajustes finos nos parâmetros dos algoritmos.

De acordo com os resultados mostrados na Seção anterior, depois de pré-processar as fontes de dados e realizar o ajuste fino das técnicas, o algoritmo SVM ajustado apresentou a melhor eficácia em ambas as fontes de dados, alcançando um valor na medida f-measure igual para 0,83, após aplicação do primeiro exame. Em outras palavras, as técnicas de SVM com ajustes finos são capazes de identificar com pelo menos 83% de eficácia que os alunos provavelmente falharão quando tiverem realizado pelo menos 25% do curso. Cabe resaltar que tais resultados não foram obtidos considerando-se a análise mais precoce, ou seja, durante as três primeiras semanas do curso em questão.

Entre os resultados obtidos, observou-se que as técnicas analisadas: Indução de Árvore de Decisão, Naïve Bayes, Máquina de Vetores de Suporte e

Redes Neurais, foram todas capazes de identificar com antecedência e alta confiabilidade, estudantes com tendência ao insucesso, podendo prestar-se para auxiliar professores, oferecendo-lhes informação relevante para ajudar em suas decisões pedagógicas. Além disso, mostrou-se também que a efetividade destas técnicas foi melhorada, sobretudo indicando em quanto, após realização de um pré-processamento nos dados, notadamente com o uso de técnica de seleção de atributos e balanceamento nos dados, bem como, com aplicação de ajuste fino nos algoritmos. No mais, a técnica SVM mostrou fornecer um desempenho estatisticamente significativo e superior às demais, predizendo com acurácia de 92% e 83%, nas modalidades a distância e presencial, respectivamente, o insucesso de estudantes, que frequentaram pelo menos 50% dos cursos.

4.4 Predição de Desempenho: Estudo III

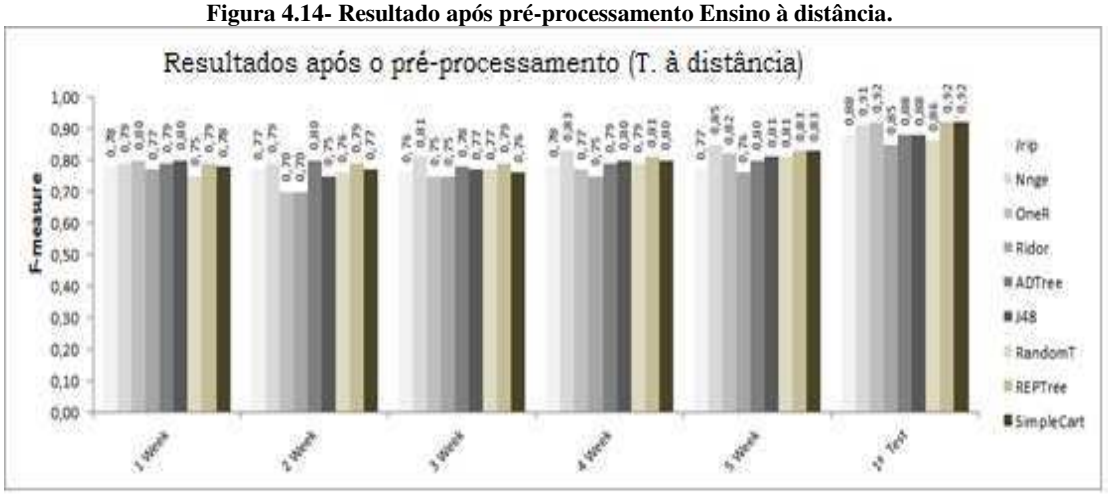
Prossegue-se aqui com um estudo focado nos comportamentos dos modelos preditivos caixa branca. No entanto, um outro aspecto foi incluído nesta pesquisa, abordando também a seguinte questão de pesquisa:

1. Existe diferença entre os desempenhos dos modelos preditivos gerados pelos algoritmos SVM e os algoritmos caixa branca, quando aplicados as mesmas fontes de dados aqui utilizadas?

Particularmente, foi também investigado o uso do algoritmo do vizinho mais próximo, k-NN: k Nearest Neighbors (Cover & Hart, 1968), além de ampliar o repertório de algoritmos baseados em árvore, incluindo mais oito que estão disponíveis no WEKA (Witten and Frank, 2005): SimpleCART (Breiman et al., 1984), ADTree (Freund & Mason, 1999), RandomTree, REPTree (Witten and Frank, 2005). A síntese desse investimento nos novos algoritmos está sumarizada nos gráficos das Figuras 4.14 e 4.15. Neles, percebe-se que em relação aos dados do ensino presencial, o algoritmo de árvore de decisão J48 continua apresentando os melhores resultados em termos de acurácia (84,04) e de f-measure (0,80). Em relação ao ensino à distância, em termos de acurácia, o algoritmo SVM só consegue as melhores taxas após os ajustes finos. Em relação à taxa de f-measure, os algoritmos OneR, REPTree e

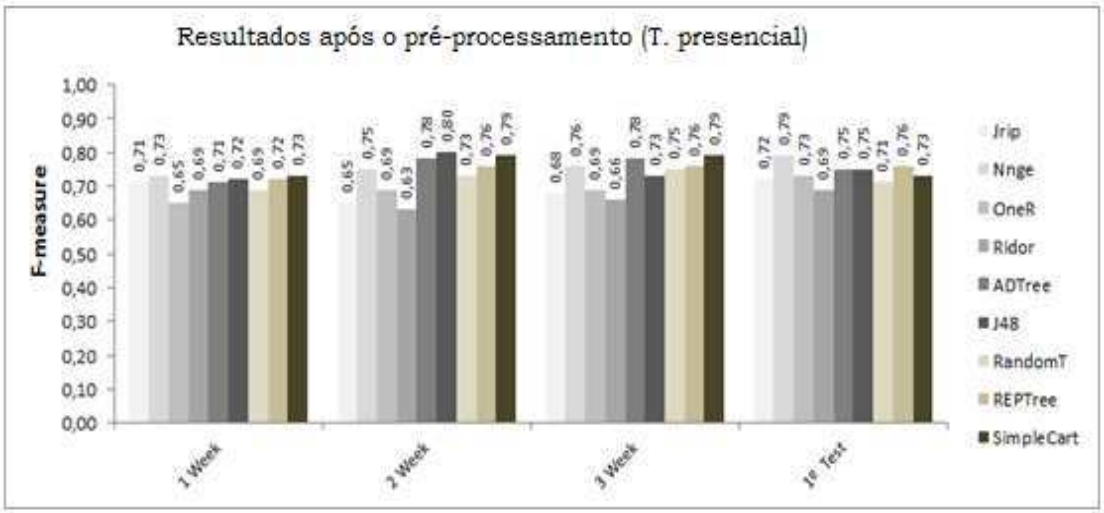
SimpleCart apresentaram a mesma taxa que o algoritmo SVM após os ajustes finos (0,92). Neste ponto, ressalta-se a importância destes algoritmos baseados em árvore se equipararem ao desempenho do SVM, pois eles são mais atraentes para serem escolhidos devido à legibilidade dos modelos, ao passo que o SVM não tem tal característica.

Figura 4.14- Resultado após pré-processamento Ensino à distância.



Fonte: Adaptada de Costa et al. (2017).

Figura 4.15 Resultado após pré-processamento Ensino Presencial



Fonte: Adaptada de Costa et al. (2017).

Outro aspecto que demanda atenção é a fase de pré-processamento, principalmente investigando outros algoritmos de seleção de atributos. Mais ainda, serão buscados novos atributos ainda não contemplados, por exemplo, buscados no fórum de discussão, tentando enriquecer ainda mais o conjunto

de atributos existentes.

Em síntese, busca-se com todo esse investimento tirar o máximo de acurácia na predição de desempenho acadêmico, então partindo-se para etapa com dois conjuntos bem classificados, dentro dos requisitos aqui perseguidos: confiabilidade, antecedência e automaticidade. Os resultados obtidos sobre o desempenho destes algoritmos caixa branca já se mostraram relevantes, motivando mais estudos em busca de melhorias, o que está retomado no Capítulo 5 desta tese.

4.5 Predição de Desempenho: Estudo IV

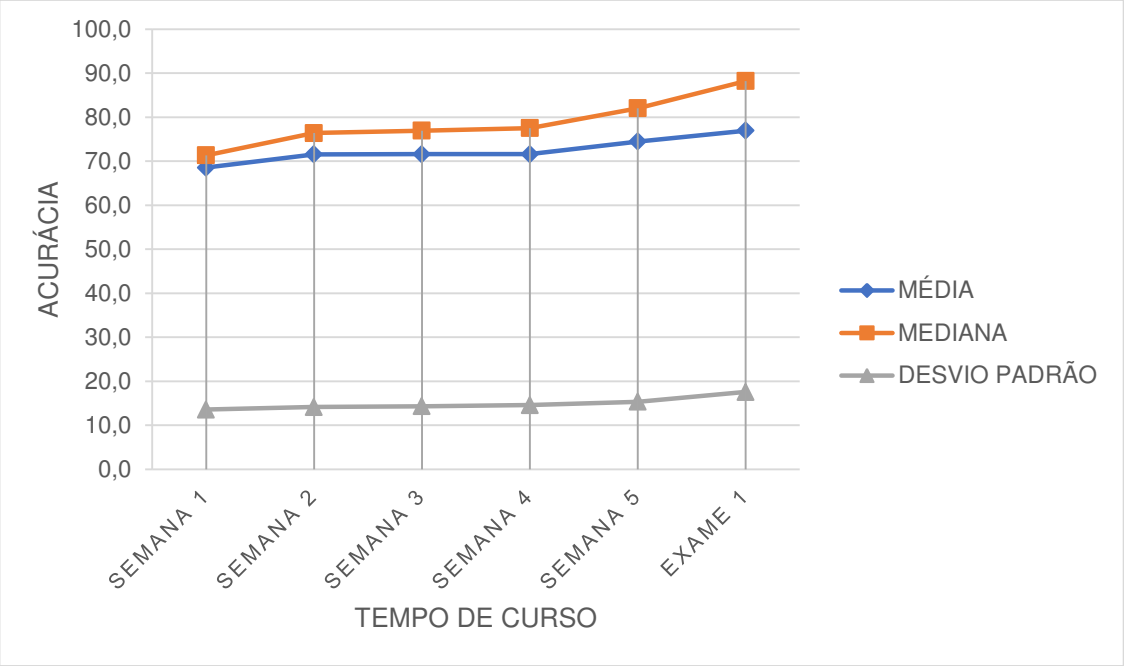
Neste último estudo desta etapa preliminar, buscou-se abordar as mesmas duas fontes de dados utilizadas nos três estudos anteriores, focado na questão da predição antecipada. Neste sentido, procurou-se aprimorar os resultados obtidos anteriormente, inclusive melhorando alguns dos resultados publicados em [Costa et al., 2017], conforme está discutido no final desta seção. Assim, os dados em apreço são:

a) EAD: contém informações sobre 262 estudantes de graduação que fizeram o curso introdutório de programação na modalidade de educação a distância, no curso de sistema de informação da Universidade Federal de Alagoas, em 2013, durante 10 semanas. Neste curso os alunos foram avaliados semanalmente de acordo com suas atividades e mais dois exames aplicados na quinta e última semana do curso. Essas atividades e exames foram aplicados por meio da Plataforma Moodle, ao invés da Plataforma TheHuxley.

b) Presencial: A segunda fonte de dados contém informações sobre 161 alunos que fizeram o curso de programação introdutória realizado no campus em nossa universidade em 2014, durante 16 semanas. Neste curso, os alunos foram avaliados semanalmente por atividades na Plataforma TheHuxley.

Pode-se observar que também há tendência de crescimento de acurácia, embora seja um crescimento mais sutil em relação ao tempo, vistos de forma mais explícita nas Tabelas 4 e 5. Um ponto importante a salientar são os níveis de acurácia superiores alcançados já na primeira semana de curso, equivalente a T1, tanto presencial quanto EAD, chegando em média próximo a 70%. Isto se deve ao fato de que nesses dados estão presentes informações de grande relevância dos alunos, como por exemplo quantidade de postagens feitas pelo aluno no ambiente online. Outro fator que contribui para a eficiência temporal destes dados é a quantidade de instâncias.

Figura 4.16 Média, Mediana e Desvio Padrão das acurácias dos 14 Algoritmos de classificação aplicados aos dados da turma da Modalidade a Distância, 2013.



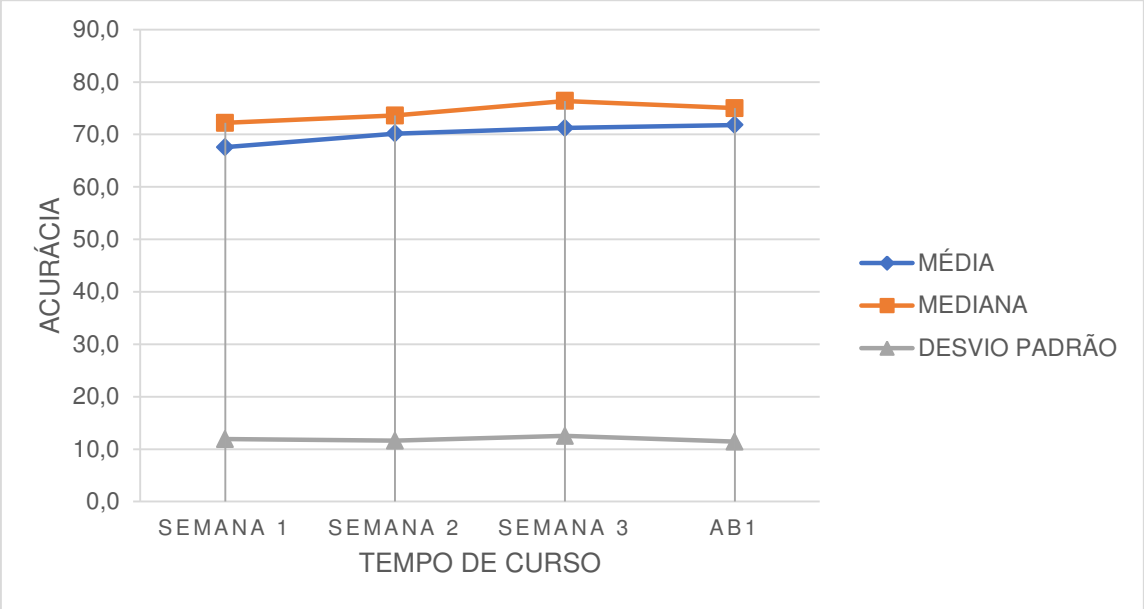
Fonte: Elaborada pelo Grupo de Pesquisa TIPS

Tabela 4 - Acurácias dos 14 Algoritmos de Classificação aplicados aos dados da Modalidade a Distância, 2013

Algoritmo	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Exame 1	MÉDIA	MEDIANA
JRip	71,3	79,8	73,6	81,5	79,8	90,4	68,1	79,8
NNge	71,3	74,2	77,3	78,1	83,1	88,8	67,5	77,3
OneR	49,4	49,4	49,4	49,4	49,4	49,4	42,4	49,4
Prism	52,8	57,9	57,9	60,7	64,0	65,7	51,3	57,9
Ridor	75,3	75,8	78,7	78,7	82,0	91,0	68,8	78,7
ADTree	75,3	80,9	77,5	84,3	86,5	89,3	70,5	80,9
J48	65,7	76,4	75,8	77,5	84,8	89,3	67,1	76,4
RandomTree	63,5	78,1	73,6	61,8	74,2	65,2	59,5	65,2
REPTree	52,8	52,8	52,8	52,8	52,8	52,8	45,3	52,8
SimpleCart	60,7	60,7	60,7	60,7	60,7	60,7	52,0	60,7
SVM - kernel:linear	85,4	87,1	86,0	84,8	87,6	89,9	74,4	86,0
SVM - kernel:polynomial	76,4	74,7	77,0	74,2	82,6	89,9	67,8	76,4
SVM - kernel:RBF	78,7	77,0	77,5	77,5	79,2	80,9	67,3	77,5
SVM - kernel:sigmoid	39,3	39,3	39,3	39,3	39,3	38,8	33,6	39,3
Naive Bayes	80,9	82,6	86,0	83,1	84,8	85,4	71,8	83,1
Rede Neural	85,4	85,4	86,0	86,5	89,3	92,7	75,0	86,0
KNN	80,9	84,8	88,2	86,5	86,0	88,2	73,5	86,0
MÉDIA	**68,5**	**71,6**	**71,6**	**71,6**	**74,5**	**77,0**		
MEDIANA	**71,3**	**76,4**	**77,0**	**77,5**	**82,0**	**88,2**		
DESVIO PADRÃO	**13,59**	**14,14**	**14,35**	**14,58**	**15,34**	**17,59**		

Uma lacuna encontrada na análise temporal da pesquisa citada é o fato de que é explorado apenas o período até a primeira avaliação bimestral, ficando desconhecidos os resultados de desempenho até o final do curso. Uma análise mais abrangente pode ser apresentada em trabalho futuro, estendendo a análise comparativa e confirmando o que acontece após a primeira avaliação bimestral: para todos os grupos de algoritmos estudados, o desempenho permanece praticamente constante após T4.

Figura 4.17- - Média, Mediana e Desvio Padrão das acurácias dos 14 Algoritmos de classificação aplicados aos dados da turma Presencial 2014.



Fonte: Elaborada pelo Grupo de Pesquisa TIPS

Tabela 5 - Acurácias dos 14 Algoritmos de Classificação aplicados aos dados Turma Presencial 2014.

Algoritmo	Semana 1	Semana 2	Semana 3	AB1	MÉDIA	MEDIANA
JRip	72,2	73,6	83,3	77,8	76,7	75,7
NNge	72,2	79,2	80,6	76,4	77,1	77,8
OneR	79,2	79,2	79,2	73,6	77,8	79,2
Prism	37,5	44,4	44,4	47,2	43,4	44,4
Ridor	72,2	72,2	76,4	75,0	74,0	73,6
ADTree	77,8	77,8	83,3	76,4	78,8	77,8
J48	73,6	81,9	77,8	73,6	76,7	75,7
RandomTree	79,2	75,0	79,2	83,3	79,2	79,2
REPTree	75,0	79,2	79,2	81,9	78,8	79,2
SimpleCart	73,6	79,2	79,2	75,0	76,7	77,1
SVM - kernel:linear	59,7	68,1	65,3	80,6	68,4	66,7
SVM - kernel:polynomial	69,4	72,2	73,6	76,4	72,9	72,9
SVM - kernel:RBF	56,9	56,9	56,9	56,9	56,9	56,9
SVM - kernel:sigmoid	44,4	44,4	44,4	44,4	44,4	44,4
Naive Bayes	61,1	62,5	61,1	68,1	63,2	61,8
Rede Neural	72,2	75,0	76,4	79,2	75,7	75,7
KNN	72,2	72,2	70,8	75,0	72,6	72,2
MÉDIA	**67,6**	**70,2**	**71,2**	**71,8**		
MEDIANA	**72,2**	**73,6**	**76,4**	**75,0**		
DESVIO PADRÃO	**11,91**	**11,60**	**12,53**	**11,43**		

4.6 Síntese

Buscou-se, neste capítulo, apresentar, primeiramente, um estudo avaliativo em técnicas de mineração de dados educacionais, objetivando-se comparar a eficácia dos algoritmos de predição capazes de identificar, em tempo hábil para potencializar intervenção pedagógica oportuna, ajudando os estudantes propensos ao insucesso. Assim, avaliou-se a eficácia de algoritmos de predição em duas fontes de dados diferentes e independentes, uma na modalidade de ensino presencial e a outra na modalidade de ensino a distância sobre as disciplinas de programação introdutória. Os resultados mostraram que as técnicas analisadas no estudo são eficazes na identificação dos estudantes propensos ao insucesso no início da disciplina. Além disso, mostrou-se também que após a realização das etapas de pré-processamento e ajustes nos parâmetros dos algoritmos, tais algoritmos analisados tiveram uma melhora em seus resultados. Ao fim do processo, o algoritmo máquina de vetor de suporte (SVM: Support Vector Machine) apresentou os melhores resultados, tanto na modalidade de ensino presencial quanto na modalidade a distância, alcançando uma taxa de f-measure de 92% e 83%, respectivamente. No entanto, na realização do estudo complementar, investindo-se em algoritmos do tipo caixa branca, resultados significativos e no mesmo patamar do que foi obtido com o algoritmo SVM, foram obtidos com os algoritmos OneR, REPTree e SimpleCart. Por fim, no estudo IV procurou-se reproduzir os experimentos nos estudos I, II e III, procurando avançar nos resultados, o que ocorreu em alguns aspectos.

5 ABORDAGEM PREDITIVA: PROCESSOS E RESULTADOS

Neste capítulo, dando continuidades aos resultados já anunciados no capítulo anterior, conclui-se a abordagem preditiva pretendida, apresentando-se uma resposta ao mencionado problema de identificação, tão cedo quanto possível, de estudantes com risco de insucesso, isto aqui mapeado no problema da predição antecipada de estudantes com propensão ao insucesso acadêmico durante a realização de uma disciplina de programação introdutória, considerando-se o objetivo de obtenção de uma informação confiável, oportuna e útil envolvida em uma solução de compromisso entre acurácia satisfatória e compreensibilidade do modelo adotado. Assim, procurou-se ao longo das seções seguintes descrever os processos envolvidos, desde a preparação dos dados até a etapa de visualização de alguns aspectos do modelo preditivo desenvolvido. Deste modo, como resposta às questões de pesquisa decorrentes do problema referido anteriormente, enfatizou-se principalmente o processo utilizado na seleção e ranqueamento de atributos, o processo utilizado na etapa preditiva, além dos aspectos relacionados à explicabilidade do modelo.

5.1 Metodologia

Na perspectiva mencionada anteriormente, a presente investigação científica buscou responder ao seguinte problema de pesquisa: Como desenvolver um modelo preditivo que permita automaticamente identificar desempenho acadêmico de estudantes em risco de insucesso, considerando-se um solução de compromisso entre acurácia satisfatória e compreensibilidade do modelo adotado, a partir de dados destes estudantes antes de iniciar o curso, e, principalmente, dados do comportamento de tais estudantes, obtidos da plataforma de aprendizagem online durante o curso? A partir deste problema macro, considerando-se mais especificidades sobre as questão do Capítulo 4, as seguintes questões de pesquisa foram elencadas para que fossem respondidas durante esta etapa conclusiva de desenvolvimento da presente

tese, quais sejam:

Questão 1: Quais atributos, identificados em um universo estabelecido envolvendo dados de antes e durante o curso, podem influenciar o insucesso dos estudantes, devendo ser selecionados como os melhores indicadores, satisfazendo a qualidade da predição de desempenho acadêmico de tais estudantes?

Questão 2: Quão eficientes são os algoritmos de classificação aplicados na predição do insucesso de estudantes de programação introdutória no momento de início do curso, baseando-se em dados prévios, ou seja os que estão associados aos estudantes antes dele iniciar a disciplina?

Questão 3. Quão eficientes são os algoritmos de classificação caixa branca, por regras e por árvores de decisão, no que se refere, respectivamente, aos atributos nos antecedents das regras e ao nível de profundidade das árvores, observados particularmente no momento de início do curso?

Questão 4: A partir de qual período no curso, consegue-se obter predição de desempenho acadêmico satisfatória, portanto, focalizando a tarefa predição antecipada de insucesso, bem como qual o comportamento dos algoritmos no momento mais cedo possível identificado e após ele?

Para responder esta questão de pesquisa, ampliou-se os estudos, já descritos no Capítulo 4, investindo-se em outros estudos exploratórios, abordando-se, principalmente, aspectos de eficiência na construção dos modelos preditivos.

Como decorrente desta Questão 4, as seguintes sub-questões particularmente serão também aqui respondidas.

Questão 4.1): Quais os algoritmos preditivos do tipo caixa branca, mas tendo um caixa preta satisfatório como referência, apresentam melhor desempenho para serem escolhidos como solução viável antes de iniciar o curso (T0)?

Questão 4.2): Quais os algoritmos preditivos do tipo caixa branca, mas tendo um caixa preta satisfatório como referência, apresentam melhor

desempenho para serem escolhidos como solução viável durante o curso, focalizando o momento mais cedo possível?

Questão 4.3): Quais os atributos se mostram relevantes para serem utilizados nos algoritmos de predição em T0?

Questão 4.4): Quais atributos se mostram relevantes para serem utilizados em predições durante o curso?

Questão 5: Como prover formas de explicação que possam permitir ao professor compreender adequadamente as informações produzidas no processo de predição aplicado, considerando os algoritmos caixa branca?

As questões seguintes foram elaboradas para auxiliar em algum aspecto envolvido principalmente na questão 4, por exemplo verificando questão de desempenho com a tarefa de seleção de atributos, mas se prestando também para a questão 5.

Questão 1: Quais atributos, identificados em um universo estabelecido envolvendo dados de antes e durante o curso, podem influenciar o insucesso dos estudantes, devendo ser selecionados como os melhores indicadores, satisfazendo a qualidade da predição de desempenho acadêmico de tais estudantes?

Questão 2: Quão eficientes são os algoritmos de classificação aplicados na predição do insucesso de estudantes de programação introdutória no momento de início do curso, baseando-se em dados prévios, ou seja os que estão associados aos estudantes antes dele iniciar a disciplina?

Questão 3. Quão eficientes são os algoritmos de classificação caixa branca, por regras e por árvores de decisão, no que se refere, respectivamente, aos atributos nos antecedents das regras e ao nível de profundidade das árvores, observados particularmente no momento de início do curso?

Ressaltou-se o uso de modelos preditivos do tipo caixa branca, visando, além de assegurar uma acurácia preditiva satisfatória, oferecer uma compreensibilidade adequada do modelo aos seus usuários. Deste modo, investiu-se em modelos preditivos baseados em árvores e em regras, mas

tendo como referência na taxa de acurácia um representante de modelo preditivo do tipo caixa preta, considerando-se o seu bom desempenho nos estudos aqui realizados, revelando resultado próximo dos algoritmos caixa branca. Além disso, definiu-se um processo para predição aplicada em diferentes momentos do curso, inclusive com uma execução inicial antes de começar o curso. Ademais, a abordagem, e seus processos, foi instanciada em dois diferentes estudos avaliativos em instituições públicas federais, um no ensino superior e o outro no ensino médio, verificando-se, inclusive aspectos de reprodutibilidade da proposta.

Fontes de Dados

Nos estudos empíricos que estão desenvolvidos a seguir, foram usados dados de diferentes fontes, já mencionadas, tais como:

SIE WEB: Tal como já mencionado, trata-se do Sistema Acadêmico Online da UFAL, disponível em https://sistemas.ufal.br/academico , o qual mantém uma base de dados acadêmicos dos estudantes de graduação da UFAL. Nesta fonte de dados, dispõe-se, dentre outros, do atributo “Nota Final”, isto sendo o dado para se obter a informação sobre se o aluno foi aprovado ou reprovado na disciplina. Portanto, este atributo, na etapa de Pré-processamento, é transformado em categórico, e, assim, são criadas 2 classes para descrevê-lo: R – Reprovado e A- Aprovado. Este atributo, por sua vez, é então posicionado no *status* de Classificador, para uso no processo de Predição e Seleção de Atributos. O conjunto de dados extraídos desta fonte foi aqui denominado de “Dados Acadêmicos”.

Huxley: Tal como já descrita, trata-se de uma plataforma online para aprendizagem e testes em Programação, disponível em https://www.thehuxley.com. Para melhor identificação, adota-se a expressão “Dados do Huxley” para significar o subconjunto de dados coletados desta fonte.

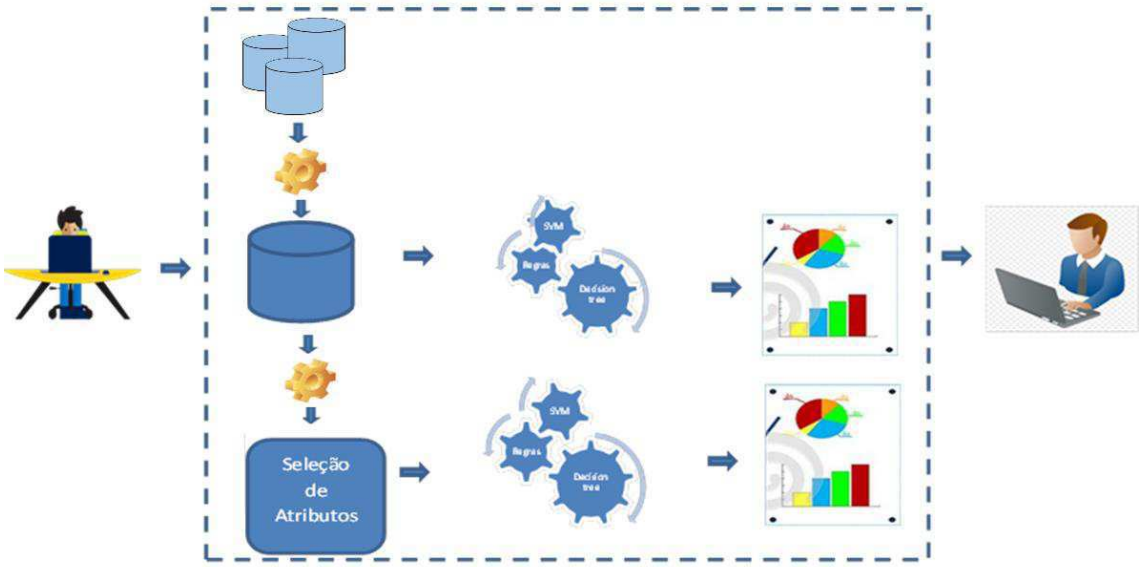
ENEM 2012: Considera-se desta fonte atributos socioeconômicos e acadêmicos, referentes aos candidatos que prestaram o ENEM - Exame

Nacional do Ensino Médio, no ano de 2012, na qual estão contidos os dados dos estudantes que ingressaram na Instituição em 2013. Tais dados estão disponíveis em http://portal.inep.gov.br/web/guest/microdados. Esta fonte de dados foi escolhida por permitir acesso a dados acadêmicos, principalmente as notas das provas, e dados socioeconômicos dos estudantes, os quais foram aqui considerados, devido, notadamente, ao fato de ter sido identificado dados desse tipo em várias pesquisas relatadas na literatura, o que, por exemplo, permitirá comparações. Como a disciplina de Programação I, alvo desta pesquisa, é uma disciplina inicial e obrigatória, todos os alunos ingressantes, obrigatoriamente devem se matricular nela, deste modo foi possível a localização dos alunos com vistas à pretendida unificação das 3 sub amostras oriendas das três fontes de dados em uso.

5.2 Arcabouço da Pesquisa e Estudo Avaliativo

O arcabouço conceitual da pesquisa ressaltando um processo com as principais etapas e fluxos envolvidos na abordagem preditiva aqui proposta está exibido na Figura 5.1, destacando desde as atividades de coleta e preparação de dados até a etapa de predição e exibição dos resultados. No entanto, uma primeira perspectiva, enfatizando aspectos mais gerais desse processo foi mostrada no Capítulo 4, na Figura 4.1. Algumas dessas etapas do processo, da seleção de atributos à predição, foram realizadas com o suporte da plataforma WEKA, mas poderia ser substituída por outra plataforma. Especificamente, conforme a Figura 5.1, ressaltadas as fontes de dados em consideração, inicia-se com a etapa que envolve mecanismos para coleta e integração das fontes de dados, prosseguindo-se ainda nesta etapa de pré-processamento, realizando operações que levem à preparação da base de dados, daí então seguindo-se no fluxo por dois caminhos para realizar a predição: um deles que executa diretamente os algoritmos de predição sobre a base de dados, sem que tenha havido seleção de atributos, já o outro caminho passa pela operação de seleção de atributos, então executando os algoritmos de predição sobre a base de dados com os atributos selecionados.

Figura 5.1- Fluxo de Execução da Abordagem preditiva



Fonte: Autora

Neste estudo avaliativo, utilizou-se a abordagem aqui proposta para oferecer uma resposta ao problema macro mencionado anteriormente neste capítulo, adotando-se uma metodologia que consiste em duas grandes fases descritas a seguir. Para este estudo especificamente, foram tratados os fatores que poderiam influenciar na eficiência das técnicas preditivas, quanto ao sucesso dos estudantes. Assim, o processo principal da metodologia proposta neste estudo foi dividido em 2 abrangentes fases: 1) Fase inicial – Comum a todos os experimentos aqui descritos, consistindo em duas partes: a. Coleta e Preparação dos Dados. b. Pré-processamento 2) Fase de experimentação – Aplicada nos 4 experimentos, sendo assim dividida em: (i) Experimento 1 – Processo de Análise da Seleção de Atributos; (ii) Experimento 2 – Processo de Análise da Eficiência dos Algoritmos de predição com diferentes parcelas de treino/teste; (iii) Experimento 3 – Processo de Análise da Eficiência dos Algoritmos de predição via árvores de decisão; (iv). Experimento 4 – Processo de análise da Eficiência dos algoritmos de predição no decorrer do curso. Além disso, particularmente, é de interesse desta pesquisa, no quesito análise de eficiência, considerar os dados existentes sobre cada aluno no momento em

que ele inicia o curso, visando analisá-los e gerar informação pretensamente valiosa. Uma motivação para esta análise preliminar se justifica pela possibilidade de que já de início poderia-se ter uma informação geral da turma, tendo uma primeira noção de alunos predispostos à reprovação, o que já daria um sinal ao professor, ajudando-o em sua atividade de tomada de decisão. Ressalta-se que apenas o Experimento 4, contido nesta pesquisa, considera dados dos alunos no momento inicial do curso.

Fase Inicial

Nesta etapa, fez-se o planejamento inicial a respeito do escopo, relevância de dados para a pesquisa, disponibilidade, procedimentos de coleta, e ainda preparação e formatação dos dados para que se tornassem adequados para a aplicação das técnicas de Seleção de Atributos e Predição.

Uma vez definidas as fontes de dados já descritas, verificou-se quais os dados de interesse de cada fonte, procedimentos para extração e unificação, além do modelo final da base de dados a ser utilizada nos estudos experimentais.

Processo de Coleta de Dados e Unificação

Nesta fase de extração dos dados, para cada uma das fontes utilizadas foram necessários procedimentos diferentes, conforme suas especidades. Por exemplo, os Dados Acadêmicos, extraídos do Sistema SIE WEB, foram disponibilizados, pela equipe de suporte da UFAL que os administra, em um arquivo no formato .csv, contendo todos os registros acadêmicos do período selecionado para estudo, no caso o 2013.2. A partir disso, foram filtradas apenas as instâncias dos alunos dos cursos de interesse, no caso, Ciência da Computação e Engenharia da Computação. Logo em seguida fazendo a filtragem na disciplina alvo desta pesquisa, no caso, Programação I, fazendo-se uso da ferramenta Microsoft Excel. Além disso, também foram necessários alguns tratamentos na base de dados para que atendesse ao objetivo da

pesquisa. Deste modo, foram descartados atributos entendidos como sem importância para o problema, tais como: endereço, e-mail, e outros, permanecendo os dados acadêmicos referentes a notas, além dos dados pessoais: Idade, Sexo, Etnia e Deficiência.

Para a extração dos dados do banco associado ao Sistema Huxley, recorreu-se à execução de operações de consultas SQL realizadas diretamente no banco de dados do sistema, gerando um arquivo .csv, com os dados de cada aluno e suas notas nas atividades realizadas durante o curso de Programação I, no período 2013.2, sendo tais operações efetuadas pela equipe responsável pela manutenção do sistema.

Quanto à extração dos dados do ENEM 2012, houve um pouco mais de dificuldade devido ao fato de a fonte de dados disponível no site do INEP encontrar-se unificada em nível nacional, portanto em um só arquivo de dados, no qual estão armazenados os dados de todos os candidatos que se inscreveram no ENEM no ano de 2012 em todo o Brasil. Assim, para resolver o problema, utilizou-se o software RStudio para a conversão do arquivo em um banco de dados relacional, permitindo assim uma consulta direta SQL buscando os alunos que estavam matriculados na disciplina Programação I, período 2013.2.

A partir da coleta das amostras específicas de dados das 3 fontes, a tarefa passou a ser a unificação em uma só base de dados. Essa tarefa foi viabilizada por meio dos campos utilizados como chave, quais sejam: número de inscrição e nome dos alunos. Para esta tarefa foram utilizados Microsoft Excel e RStudio.

Pré-processamento

A partir dos dados unificados, para que fosse possível aplicar os algoritmos de Predição, ou mesmo de Seleção de Atributos, foram necessárias alguns ajustes na base de dados, os quais fazem parte da etapa de pré-processamento de dados, quais sejam: Limpeza de dados inconsistentes, integração, categorização, balanceamento de dados e transformação de

variáveis são algumas das alterações necessárias. Tais alterações foram realizadas do seguinte modo: uma parte foi efetuada utilizando o Microsoft Excel e a outra parte usando o pacote de software WEKA.

Primeiramente, foram removidos os atributos que não se mostraram relevantes, por exemplo, dados sem integridade ou ainda atributos com instâncias sem resposta. Portanto, dados associados a tais atributos poderiam comprometer o resultado dos experimentos por não agregar informação.

O segundo passo, considerado relevante para os algoritmos de predição e seleção de atributos, foi a normalização dos atributos que continham valores numéricos. No caso, como os dados numéricos representam atributos de grandezas distintas, portanto de escalas numéricas diferentes, para que não houvesse um tratamento diferenciado pelos algoritmos, onde possivelmente os atributos com maior valor numérico seriam privilegiados, foi utilizada uma operação de normalização sobre todos os atributos numéricos, por meio da métrica **Standard Score**, na qual x é o valor numérico do dado ou amostra, μ é a média amostral e σ o desvio padrão amostral.

$$Standard\ Score = \ Z\ score = \frac{x - \mu}{\sigma}\ . \quad (6)$$

Outra transformação necessária para os experimentos foi sobre o atributo escolhido como classificador, atributo este que foi utilizado pelo algoritmo de classificação para a aprendizagem e criação do modelo, além de também ter sido usado no teste do modelo nos dados e obtenção das métricas de desempenho do algoritmo. No caso, como o propósito é a predição de sucesso acadêmico, verficando-se a tendência de se o aluno será ou não aprovado, o atributo que escolhido como classificador neste caso foi o “Conceito”, o qual diz respeito ao conceito final do aluno e foi, deste modo, tratado como categórico considerndo-se duas categorias: A = Aprovado e R = Reprovado. Assim, tem-se dessa forma todas as instâncias da base de dados dividida em 2 classes, a dos alunos aprovados e a dos reprovados.

Em seguida, um procedimento importante e utilizado nos estudos de

predição, foi o balanceamento de classes, consistindo na equilibração das classes contidas na base de dados, a fim de que ficassem igualmente distribuídas em termos do número de instâncias. O balanceamento foi feito por meio de 2 procedimentos bem relatados na literatura: *oversampling* e *undersampling* (Faceli et al., 2011). *Oversampling* é utilizado quando, para o balanceamento, são criadas sinteticamente novas instâncias para a classe que contém menor número de instâncias fique com número igual à classe de maior número. Já o *undersampling* consiste na redução do número de instâncias, através de uma escolha aleatória, da classe de maior número de forma que fique com mesmo quantitativo de instâncias da classe de menor número. Para lidar com problema de desbalanceamento de classes, utilizou-se uma técnica bastante aplicada para tratar oversampling, que é o SMOTE - Synthetic Minority Over-sampling Technique (CHAWLA et al., 2002), a qual se presta a ajustar a frequencia relativa entre classe majoritárias e minoritárias. O WEKA possui uma implementação do algoritmo SMOTE, o qual foi utilizado neste trabalho.

Vale ressaltar que, apenas para execução de um dos algoritmos de predição utilizados, no caso, o Prism, foi necessária a transformação de todos os atributos numéricos em categóricos, pois ele aceita apenas dados categóricos como entrada. Assim, para realizar tal procedimento, utilizou-se o filtro NumericToNominal, existente no pacote WEKA.

Após a operação de unificação das fontes, obteve-se uma base de dados resultante com 59 atributos no total, e 66 instâncias de alunos. Na Tabela 6, descreve-se os atributos coletados e mantidos de cada fonte de dados coletada. Ressalta-se que os dados oriundos do Huxley foram considerados apenas no Experimento 4, assim não fazendo parte dos experimentos 1, 2 e 3.

Tabela 6 - Conjunto de Dados Coletados

Fonte de Dados	Atributos
Acadêmicos – SIEWEB (10)	AV1; AV2; Reavaliação; Prova Final; Média Final; Conceito; Idade; Sexo; Etnia; Deficiência.
Huxley (6)	Notas da Atividade 1, Notas da Atividade 2, Notas da Atividade 3, Notas da Atividade 4, Notas da Atividade 5, Notas da Atividade 6.
ENEM (43)	Idade; Sexo; Etnia; Deficiência; Município de Residência; UF de Residência; Nota na prova de Linguagens e Códigos (ENEM); Nota na prova de Ciências Humanas (ENEM); Nota na prova de Ciências da Natureza (ENEM); Nota na prova de Matemática (ENEM); Nota na prova de Redação (ENEM); Nota geral na prova do ENEM; Com quantas pessoas mora; Nível de escolaridade do Pai; Nível de escolaridade da Mãe; Renda familiar; Situação da residência onde mora; Zona onde está localizada a sua residência; Fez ENEM para testar conhecimentos; Fez ENEM para obter uma bolsa de estudos; Quantos anos levou para a conclusão do ensino fundamental; Deixou de estudar durante o ensino fundamental; Tipo de escola em que cursou o ensino fundamental; Quantos anos levou para a conclusão do ensino médio; Deixou de estudar durante o ensino médio; Tipo de escola em que cursou o ensino médio; Cursou o programa de Educação de Jovens e Adultos; Cursou o ensino regular; Pretende aderir ao FIES; Pretende aderir ao PROUNI; Pretende aderir bolsa de estudos da instituição; Pretende aderir bolsa de estudos da empresa onde trabalha; Possui TV; Possui DVD/vídeo cassete em casa; Possui rádio em casa; Possui computador; Possui automóvel; Possui máquina de lavar; Possui geladeira; Possui freezer; Possui telefone fixo; Possui telefone celular; Possui acesso à internet em casa; Possui tv por assinatura; Possui aspirador de pó; Possui empregada mensalista; Quantos Banheiros possui em casa.

Experimento 1: Processo de Análise de Seleção de Atributos.

O primeiro experimento com o uso deste processo investiu na seleção de atributos, verificando aqueles mais influenciam no sucesso dos estudantes. Para tanto, foi definido um processo com abordagem um pouco similar à metodologia utilizada por MARQUEZ-VERA et al (2013), utilizando-se dos seus mesmos 10 algoritmos de seleção de atributos, os quais estão descritos em WITTEN E HALL (2011), e se encontram sumarizados na Tabela 6. No entanto, no processo aqui proposto se adota um critério de escolha diferente, baseado em votação, no qual aqueles atributos que foram considerados como relevantes por 6 ou mais algoritmos de seleção, dentre os 10, ou seja, mais de 50%, são considerados como influentes no sucesso dos estudantes. Tal processo permite que os atributos analisados sejam avaliados por mais de um algoritmo de seleção, apontando dessa forma aqueles que foram selecionados com maior frequência, evitando assim o risco de direcionar o resultado da

seleção a apenas um algoritmo. Conforme visto no Capítulo 2, existem diversas maneiras de se selecionar atributos, inclusive existem vários algoritmos já implementados no WEKA.

Tabela 7- Descrição dos Algoritmos de Seleção de Atributos adotados na pesquisa e implementados no WEKA

Algoritmo	Função	Categoria
CfsSubsetEval	Considera o valor preditivo de cada atributo individualmente, juntamente com o grau de redundância entre eles	Filtro
ChiSquared-AttributeEval.	Calcular qui-quadrado estatístico de cada atributo em relação à classe	Filtro
Consistency-SubsetEval	Projeta um grupo de treino dentro do conjunto de atributos e mede a consistência nos valores de classe	Filtro
Filtered-AttributeEval	Aplica um avaliador de atributo aos dados filtrados. Versão de atributo único do filtro baseadas em subconjunto	Filtro
FilteredSubsetEval	Aplica um filtro aos dados de treinamento antes que a seleção de atributos seja executada	Filtro
GainRatio-AttributeEval	Avalia atributos medindo sua razão de ganho em relação à classe.	Filtro
InfoGain-AttributeEval	avalia atributos medindo seu ganho de informação com relação à classe.	Filtro
OneRAttributeEval	Usa a medida de precisão (acurácia) simples adotada pelo classificador OneR	Filtro
ReliefFAttributeEval	Baseado em instância: faz amostragem de instâncias aleatoriamente e verifica instâncias vizinhas das mesmas e diferentes classes	Filtro
SymmetricalUncertAttributeEval	Avalia atributo com base na incerteza simétrica	Filtro

Experimento 2: Processo de Análise da Eficiência dos Algoritmos de predição com diferentes parcelas de treino/teste.

O objetivo do experimento 2 foi analisar a eficiência de cada algoritmo de classificação quando utilizados na predição de sucesso dos estudantes. Assim, a metodologia de medição da eficiência neste experimento pretendeu mostrar uma análise de quais algoritmos possuem melhor desempenho, ou acurácia, com menos dados de treino. Para isso, foram utilizados 14 algoritmos de classificação, dentre eles 10 do tipo Caixa Branca e 4 Caixa Preta, para realizar a classificação dos estudantes, utilizando, de forma progressiva, uma parcela de 10% a 90% do total de instâncias da base para treino, aprendizagem e criação do modelo, e respectivamente de 90% a 10% da base para teste e classificação. Os experimentos do 2 ao 4, foram realizados simultaneamente utilizando os dados da base sem alterações após o pré-processamento e utilizando a base de dados após processo de seleção de atributos, onde apenas os atributos mais relevantes foram mantidos, com o objetivo de comparar os desempenhos. Ressalta-se também que nos experimentos 2 e 3, foram utilizados apenas os atributos dos alunos condizentes com o seu momento de entrada no curso, não considerando, por exemplo, atributos como “Nota da Atividade ‘” ou “Nota da Prova Final”, simulando desta forma uma situação de predição real, onde o professor dispõe apenas de dados passados dos alunos, como os socioeconômicos e notas do ENEM, por exemplo. No Experimento 4, por outro lado, as notas das atividades realizadas pelos alunos durante o curso são consideradas, porém inseridas quinzenalmente permitindo uma análise temporal.

No processo de Classificação realizado nos experimentos de 2 a 4 foram utilizados os mesmos algoritmos de regras e árvore de decisão apresentados também por Marquez-Vera et al. (2013), e foram adicionados 4 algoritmos do tipo “caixa preta” conhecidos: SVM, Naive Bayes, Multilayer Perceptron e k-NN. A relação dos algoritmos utilizados e suas respectivas características descritas com maior riqueza de detalhes em Witten e Hall (2011) e me Frank et al. (2016), estando sumarizados na Tabela 7. As quatro métricas selecionadas

para apresentação dos resultados, foram as mesmas que as usadas em Marquez- Vera et al. (2013), quais sejam: True Positive Rate (TPR), True Negative Rate (TNR), Accuracy (Acc) e Geometric Mean (GM), mas a métrica Acurácia foi aqui tratada de forma predominante.

Tabela 8 - Algoritmos de Classificação do WEKA abordados

Nome do algoritmo implementado no WEKA	Função	Categoria	Método de Aprendizagem	Tipo
JRip	Implementa o algoritmo RIPPER: *Repeated Incremental Pruning to Produce Error Reduction*, incluindo a otimização global heurística do conjunto de regras.	Regras	Baseado em Busca	Caixa Branca
NNge	*nearest-neighbor method*, é um método de vizinho mais próximo para gerar regras usando exemplares generalizados não aninhados			
OneR	Algoritmo classificador One Rule, que gera regras baseadas em um único atributo.			
Prism	implementa o algoritmo de cobertura elementar para regras			
Ridor	*ripple-down rule learner* (Ridor) aprende regras com exceções gerando a regra padrão, usando a redução incremental de erro reduzido para localizar exceções com a menor taxa de erro, encontrando as melhores exceções para cada exceção e iterando.			
ADTree	*Alternating Decision Trees*, constrói uma árvore de decisão alternada para problemas de duas classes usando *boosting*	Árvore de Decisão		
J48	Baseado no algoritmo C4.5 que induz árvore de decisão. Implementa o C4.5 revisão 8.			
RandomTree	constroi uma árvore que considera um dado número de características aleatórias a cada nó.			
REPTree	*Reduced-Error Pruning*. constrói uma árvore de decisão			

	ou regressão usando a redução de ganho / variação de informação e remove-a usando a podagem de erro reduzido.			
SimpleCart	árvore de decisão para classificação que emprega a estratégia de remoção de complexidade de custo mínimo CART (*classification and regression tree*)			
LibSVM	biblioteca que inclui classificadores do tipo *wrapper* que permitem implementações de máquinas de vetores de suporte (SVM) e regressão logística de terceiros para o WEKA..	SVM	Baseado em Otimização	Caixa Preta
Naive Bayes	implementa o classificador probabilístico Naïve Bayes padrão	Naive Bayes	Probabilístico	
MultilayerPerceptron	é uma rede neural que treina usando "retropropagação" (*backpropagation*)	Rede Neural	Baseado em Otimização	
iBk	*k-nearest-neighbor* (k-NN), é um classificador de k-vizinhos mais próximos.	k-NN	Baseado em Distância	

Experimento 3: Processo de Análise da Eficiência dos Algoritmos de Classificação por Árvore de Decisão.

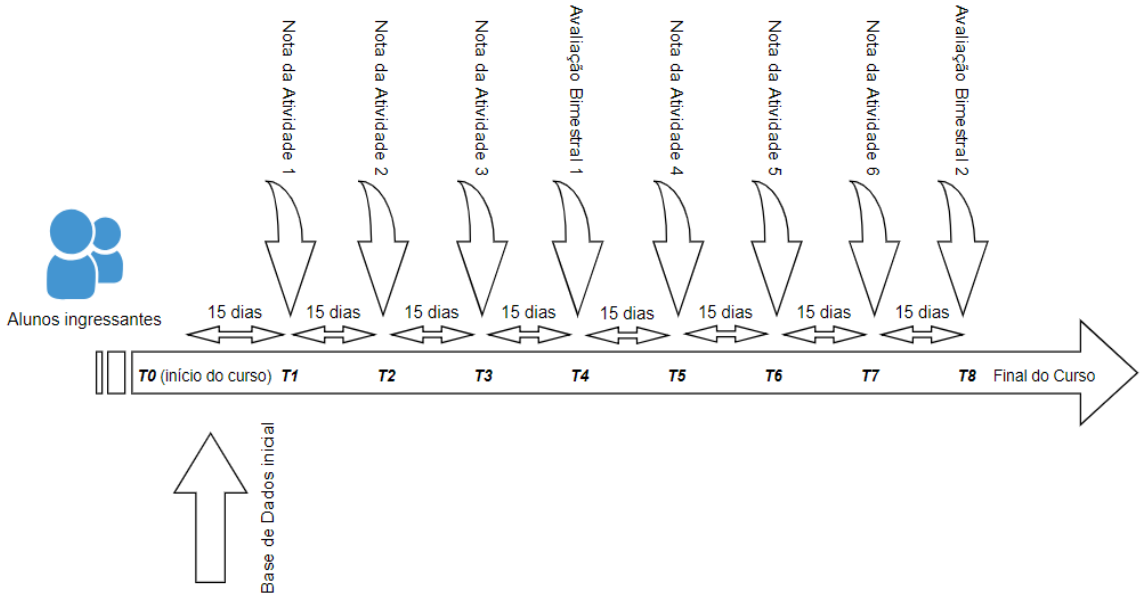
Neste experimento, o foco está voltado à eficiência dos algoritmos de Árvore de Decisão. Conforme proposto inicialmente, o objetivo é analisar os 5 algoritmos de árvore de decisão, ADTree, J48, RandomTree, REPTree, SimpleCart, verficando como eles se comportam quando possuem alterações, podas, nas árvores criadas nos modelos. Como a maioria das árvores apresentadas como resultado do processo de aprendizagem tinham aproximadamente altura igual a 5, optou-se por medir a eficiência de cada algoritmo com árvores reduzidas às alturas: 1, 2, 3, 4, 5, com podas, e também sem podas, ou seja, utilizando a árvore com o tamanho padrão apresentado no modelo de cada algoritmo. Nas opções de teste, no pacote WEKA, foi utilizado método Validação Cruzada, com 10 folds.

Experimento 4: Processo de análise da Eficiência dos algoritmos de classificação no decorrer do curso.

No experimento 4, tem-se uma análise de comportamento dos algoritmos preditivos antes de iniciar o curso, bem como, principalmente, uma análise de eficiência temporal relacionada a um dos objetivos propostos: mensurar a eficiência de cada um dos algoritmos de classificação quando, no decorrer do curso, são inseridos novos dados a respeito dos alunos, isto é, especificamente as notas dos alunos nas atividades de aprendizagem de programação realizadas na plataforma Huxley. Assim, isto envolve no processo do experimento um fluxo que conduz a duas situações diferentes, uma sem envolver seleção de atributos e outra com seleção de atributos.

Neste experimento, busca-se comparar a acurácia de cada algoritmo quando são incluídos como dados de entrada apenas dados do aluno obtidos no momento T_0, momento no qual o aluno inicia a disciplina. Neste momento, pretende-se ter um primeiro nível de informação sobre a turma, baseando-se em atributos e dados previamente conhecidos e disponíveis, incuindo dados acadêmicos, socioeconômicos e pessoais, porém no decorrer do curso mais dados são obtidos sobre o aluno, que são obtidos com base nas notas de desempenho de cada aluno nas atividades avaliativas. Deste modo, pode-se investigar melhor um esperado aumento da acurácia na predição dos algoritmos com a inserção dos dados das atividades. Na amostra de dados aqui considerada, extraída da plataforma Huxley, foram aplicadas atividades quinzenais. Na Figura 5.6 está ilustrado em um diagrama temporal, em quais momentos são inseridas as novas porções de dados contendo as notas dos alunos nas atividades de programação. Vale ressaltar que para a execução dos algoritmos de classificação no WEKA, nas opções de teste, também foi utilizado o método Validação Cruzada, com 10 folds, assim como no experimento 3, parâmetro também utilizado na maioria dos trabalhos de referência.

Figura 5.2 - Diagrama da análise temporal com dados das Notas dos alunos no Huxley.



Fonte: Grupo de Pesquisa TIPS

5.3 Resultados e Discussões

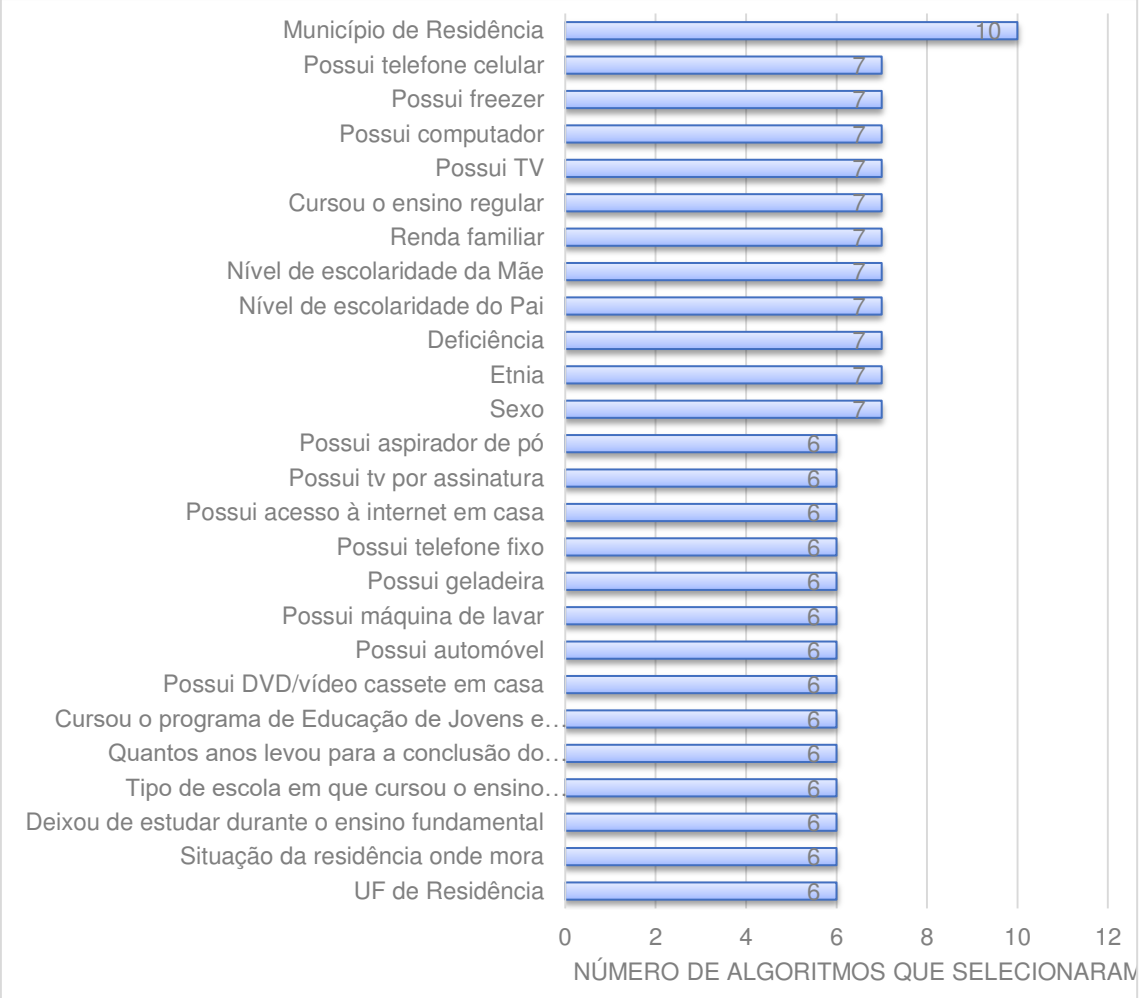
Nesta Seção são apresentados os resultados dos experimentos após a execução de todos os processos, conforme metodologia aqui proposta.

Questão 1: Quais atributos, identificados em um universo estabelecido envolvendo dados de antes e durante o curso, podem influenciar o insucesso dos estudantes, devendo ser selecionados como os melhores indicadores, satisfazendo a qualidade da predição de desempenho acadêmico de tais estudantes?

Esta questão foi abordada pelo processo de votação aqui proposto para selecionar os atributos mais relevantes, o qual consiste em executar os 10 algoritmos de seleção de atributos, selecionando 10 subconjuntos contendo os atributos apontados como relevantes por cada algoritmo. Em seguida faz-se a análise de frequência de cada atributo dentro dos subconjuntos, identificando os atributos vencedores, isto é, aqueles atributos que foram apontados como relevantes por 6 ou mais algoritmos, isto é, tendo frequência maior ou igual 6, sendo assim considerados significativamente relevantes para o estudo. Considera-se, portanto, que estes atributos, por serem mais relevantes para a predição do sucesso do estudante, também influenciam mais fortemente na

eficiência da predição dos algoritmos classificadores. A Figura 5.3 mostra graficamente os atributos mais relevantes, já exibindo informação de ranqueamento.

Figura 5.3 - Atributos considerados como mais relevantes pelos algoritmos de Seleção de Atributos.



Fonte: Autor

Analisando-se os resultados deste experimento, observa-se imediatamente a informação relativa ao atributo que se mostrou mais relevante dentre os que foram considerados, qual seja: Município de Residência. Nota-se que este atributo foi selecionado como relevante por todos os 10 algoritmos de seleção utilizados, permitindo a constatação de que há uma correlação forte entre a aprovação dos alunos e o município onde aquele aluno reside. Sobre esse resultado, pouco intuitivo, talvez uma explicação para ele esteja relacionada ao fato da necessidade do aluno em se deslocar muitas horas por

dia de sua residência até a instituição, isso podendo significar tempo perdido.

Os resultados mostram também que alguns dos principais atributos socioeconômicos, como por exemplo “Possui Celular”, “Possui Computador” ou “Renda Familiar”, também aparecem dentre os mais relevantes para o sucesso acadêmico. Uma explicação para esses resultado, possivelmente está relacionada a poder aquisitivo, ou seja, entendendo que a classe social do aluno influencia no desempenho acadêmico na disciplina de programação.

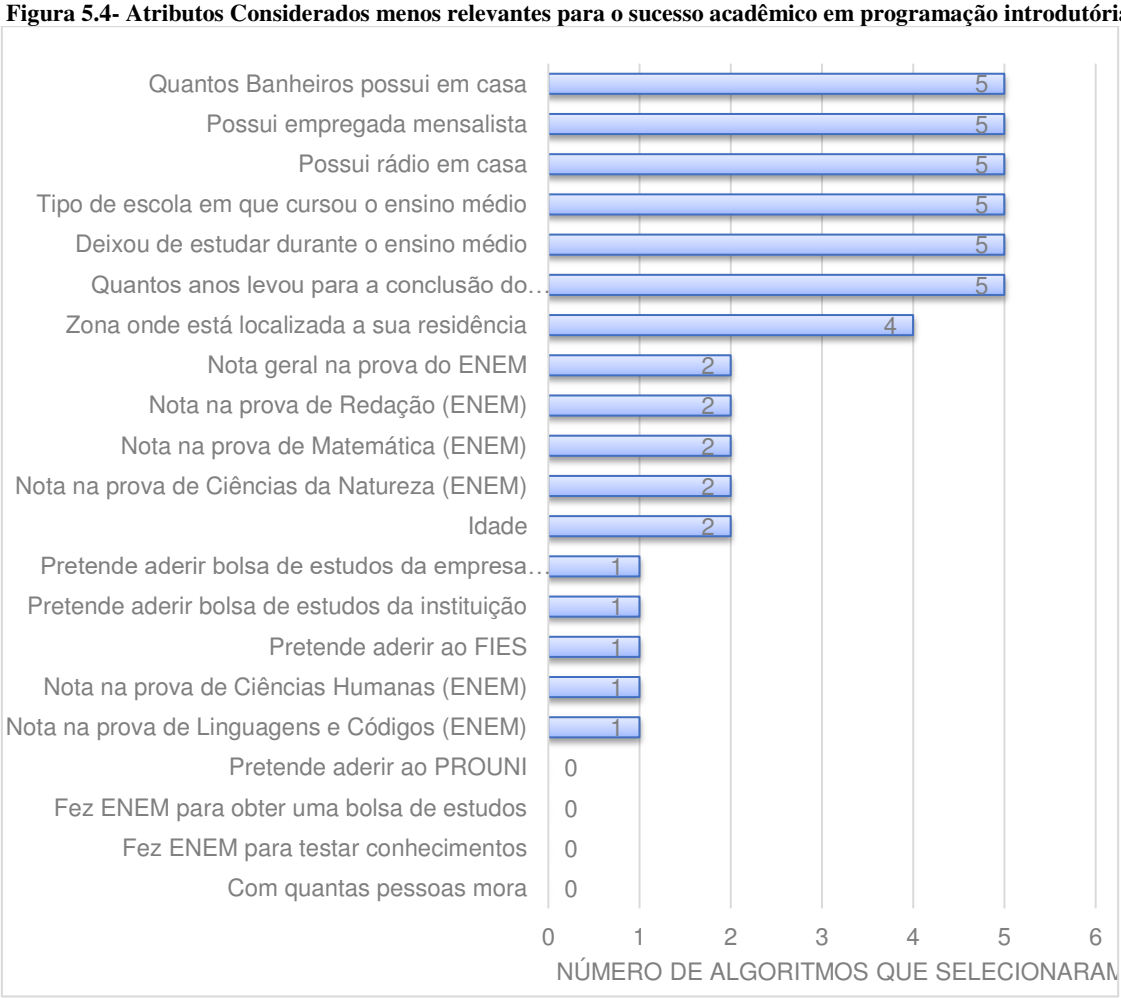
Dentre os atributos considerados relevantes em nível intermediário, podemos considerar como sendo aqueles que foram apontados por exatamente 6 algoritmos, encontramos dois que estão conceitualmente relacionados por representarem o acesso à informação do aluno, são eles: “Possui acesso à internet” e “Possui TV por assinatura”. Nota-se que o acesso à informação é apontado como um fator relevante para o sucesso acadêmico do aluno na disciplina. Vale ressaltar que no Experimento 1 apenas foram utilizados como dados de entrada os atributos dos alunos relacionados ao momento em que ele inicia o curso, ou seja, dados passados dos alunos.

A Figura 5.4, por sua vez, relaciona o ranking dos atributos menos significativos para a aprovação dos alunos em Programação I. Vemos que as pontuações dos alunos na Prova do ENEM não são consideradas relevantes, pois todos os atributos de nota no ENEM se encontram com frequência menor ou igual a 5 nas seleções. Este é um resultado curioso e que de certa forma um fator surpresa apresentado por este experimento, uma vez que é esperado que alunos que obtiveram boas notas na prova do ENEM sejam mais preparados e consequentemente sejam mais propensos à aprovação. Além disso, os 4 atributos considerados como menos relevantes em igual peso, ou seja, não foram apontados como relevantes por nenhum dos 10 algoritmos, estão relacionados com a intenção do aluno ao fazer o ENEM, intenção de obter bolsa de estudos e com quantas pessoas o aluno reside.

Em resposta à questão de pesquisa 1, conclui-se que atributos relativos à localização da residência do aluno, juntamente com os atributos relativos à classe social, direta ou indiretamente, além da questão acesso à informação, foram as características apresentadas como mais relevantes para a eficiência na predição do sucesso acadêmico de alunos de programação. Tais

descobertas não são de fácil aceitação, o que motivou uma investigação mais cuidadosa do conjunto de dados com relação a este aspecto.

Figura 5.4- Atributos Considerados menos relevantes para o sucesso acadêmico em programação introdutória



Fonte: Autor

Questão 2. Quão eficientes são os algoritmos de classificação aplicados na predição do insucesso de estudantes de programação introdutória no momento de início do curso, baseando-se em dados prévios, ou seja os que estão associados aos estudantes antes dele iniciar a disciplina?

Neste experimento 2 o objetivo é responder à questão proposta sobre a eficiência dos algoritmos de classificação, quando utilizamos menos dados para treino e mais dados para teste, e vice-versa, no processo de classificação para predição. A métrica principal utilizada para a comparação da eficiência dos algoritmos de classificação foi a Acurácia, utilizada na maioria dos trabalhos de referência da área. A Acurácia se caracteriza por mensurar, para

este caso especificamente, quantos alunos foram classificados corretamente, sejam eles aprovados ou reprovados, sobre o número total de alunos utilizados no teste do modelo criado pelo algoritmo. Como o principal interesse neste estudo é identificar aqueles alunos que correm risco de ser reprovados, a métrica Recall, também conhecida como Sensibilidade ou revocação, representa uma medida importante para o estudo pois direciona à medição do nível de acerto relativo a uma das classes, que neste caso seria a classe dos alunos Reprovados, em outras palavras, trata da razão entre os elementos TP e os (TP + FN), onde TP + FN representam todos os alunos que realmente estão em direção ao fracasso.

Em primeiro lugar, foi realizada uma análise geral com todos os 14 algoritmos de classificação, mensurando média, mediana e desvio padrão para cada uma das 9 execuções, ou seja, a classificação foi executada com cada algoritmo segmentando o total de instâncias 10% para treino e 90% para teste, 20% para treino e 80% para teste, e assim sucessivamente até chegar a 90% de treino para 10% apenas para teste. Vale lembrar que neste experimento e também no experimento 3, são considerados apenas os atributos do aluno disponíveis no instante T_0, ou seja, no momento de início do curso: dados socioeconômicos e pessoais, extraídos da fonte de dados ENEM.

Nas Tabelas a seguir, estão listados todos os resultados das acurácias individuais de cada algoritmo para o experimento 2 com e sem seleção de atributos. A visualização em números facilita a interpretação das tendências de cada algoritmo individualmente. Na Tabela 8 encontram-se os resultados das acurácias quando utilizados todos os atributos da base de dados principal, já na Tabela 9, tem-se os resultados das acurácias quando apenas os atributos selecionados através do experimento 1, considerados como mais relevantes, são utilizados. Estão destacados em ambas tabelas os valores máximos e mínimos, em linha e em coluna. Observa-se uma concentração dos piores resultados de acurácias, em vermelho, nas execuções com 80% e 90% para os algoritmos de Árvores de Decisão e também para os do tipo Caixa Preta. Pode-se constatar também que o algoritmo que apresentou pior eficiência geral foi o Prism, chegando a uma média de 22% dentre todos os resultados obtidos em suas execuções com as 9 parcelas de instâncias para treino. É importante

ressaltar que o Prism possui algumas características peculiares observadas através da matriz de confusão gerada nos experimentos: 1) Tal algoritmo sofre muito com alta dimensionalidade, ou seja, não lida muito bem com muitos atributos; 2) Possui a limitação de só trabalhar com dados categóricos, o que torna necessária a discretização dos dados numéricos de entrada antes de sua execução. 3) Sua acurácia é comprometida pelo fato de que ele deixa algumas de fora durante a sua execução, ou seja, instâncias muito próximas ao limiar de classificação de seu modelo não são classificadas nem como aprovado nem reprovado, o que consequentemente conta negativamente para o cálculo de acurácia, uma vez que são considerados com instâncias para o total de instâncias de entrada, porém não são consideradas nem como TP nem TN. A Discretização dos dados para permitir a execução deste algoritmo foi realizada diretamente no WEKA a partir do filtro já implementado NumericToDecimal.

Tabela 9 - Acurácia dos Algoritmos de Classificação com parcelas de treino de 10% a 90%

Algoritmo	10%	20%	30%	40%	50%	60%	70%	80%	90%	**MÉDIA**	**MEDIANA**
JRip	50,8	48,1	**47,9**	53,7	58,8	51,9	60,0	64,3	**71,4**	**56,3**	**53,7**
NNge	55,7	51,9	50,0	53,7	58,8	**66,7**	40,0	35,7	**28,6**	**49,0**	**51,9**
OneR	50,8	51,9	54,2	53,7	50,0	**44,4**	45,0	57,1	**71,4**	**53,2**	**51,9**
Prism	34,4	**37,0**	14,6	14,6	29,4	22,2	**10,0**	21,4	14,3	**22,0**	**21,4**
Ridor	47,5	50,0	50,0	**46,3**	47,1	51,9	**75,0**	50,0	57,1	**52,8**	**50,0**
ADTree	50,8	46,3	48,8	51,2	67,6	48,1	45,0	**35,7**	**71,4**	**51,7**	**48,8**
J48	50,8	48,1	43,8	48,8	44,1	48,1	**50,0**	50,0	**42,9**	**47,4**	**48,1**
RandomTree	59,0	42,6	50,0	51,2	50,0	**37,0**	65,0	57,1	**71,4**	**53,7**	**51,2**
REPTree	47,5	48,1	**52,1**	46,3	47,1	48,1	45,0	**42,9**	**42,9**	**46,7**	**47,1**
SimpleCart	47,5	57,4	52,1	53,7	50,0	48,1	45,0	**64,3**	**42,9**	**51,2**	**50,0**
SVM - kernel:linear	50,8	57,4	50,0	53,7	55,9	**59,3**	45,0	50,0	**14,3**	**48,5**	**50,8**
SVM - kernel:polynomial	52,5	57,4	52,1	53,7	55,9	**59,3**	55,0	42,9	**14,3**	**49,2**	**53,7**
SVM - kernel:RBF	47,5	48,1	47,9	53,7	58,8	**59,3**	50,0	50,0	**28,6**	**49,3**	**50,0**
SVM - kernel:sigmoid	47,5	48,1	47,9	46,3	47,1	48,1	45,0	**50,0**	**42,9**	**47,0**	**47,5**
Naive Bayes	52,5	50,0	50,0	**56,1**	52,9	55,6	45,0	50,0	**42,9**	**50,5**	**50,0**
Rede Neural	50,8	46,3	**56,3**	53,7	55,9	55,6	45,0	35,7	**0,0**	**44,3**	**50,8**
KNN	50,8	**38,8**	45,8	43,9	**64,7**	55,6	55,0	50,0	42,9	**49,7**	**50,0**
MÉDIA	**49,8**	**48,7**	**47,8**	**49,1**	**52,6**	**50,5**	**48,2**	**47,5**	**41,2**		
MEDIANA	**50,8**	**48,1**	**50,0**	**53,7**	**52,9**	**51,9**	**45,0**	**50,0**	**42,9**		
DESVIO PADRÃO	**5,03**	**5,78**	**9,07**	**9,56**	**8,82**	**10,05**	**13,34**	**10,99**	**22,52**		

Analisando os resultados da Tabela 9, onde foram considerados apenas os atributos mais relevantes, observa-se que a tendência de redução de

acurácia com o aumento de percentual de dados para treino persiste, confirmando o resultado anterior. Neste caso o decréscimo foi ainda mais acentuado, chegando a um valor médio geral próximo aos 40%.

Pode-se então responder à questão de pesquisa 2 a partir desta análise geral que, independente do algoritmo, seu desempenho tende a ser menor quando sujeitos a parcelas grandes de treino, como 80% ou 90%, e parcelas reduzidas de teste.

Outras conclusões podem ser obtidas através de uma análise mais precisa de eficiência na Tabela 9:

a) Os algoritmos que apresentaram maior eficiência foram NNge e SVM – kernel: linear, chegando a 61,8% de acurácia a 50% de dados de treino.

b) As maiores acurácias gerais foram alcançadas pelo Ridor e RandomTree, 71,4%. Em outras palavras, estes seriam os algoritmos mais indicados para a utilização como preditores, mesmo em casos onde há diferença significativa entre o percentual de instâncias para treino e teste. O Ridos ainda se mostra superior para a aplicação pois além de apresentar maior acurácia no geral, também alcança a maior média quando submetido à variações de instâncias de treino e teste.

c) A Seleção de atributos elevou a menor acurácia média, apresentada pelo Prism, de 22% para 38,6%, comparando-se as Tabelas 8 e 9. Este comportamento também foi apresentado nos resultados da pesquisa de Marquez-Vera et al (2013), na qual as piores acurácias apresentadas tiveram um acréscimo significativo após a seleção de atributos.

d) As Acurácias máximas e superiores à média não foram afetadas significativamente pela Seleção de Atributos neste caso, resultado semelhante ao apresentado em Marquez-Vera. Alguns inclusive apresentaram acurácia inferior após a seleção.

e) A Proximidade entre os valores Média e Mediana, tanto da tabela quanto dos gráficos, em geral, mostra que existem poucos

valores discrepantes ou concentrações de amostras de resultados para mais ou para menos, o que torna segura a utilização da média para as conclusões das tendências apresentadas.

Tabela 10- Acurácia dos Algoritmos de Classificação com parcelas de treino de 10% a 90% - com Seleção de Atributos

Algoritmo	10%	20%	30%	40%	50%	60%	70%	80%	90%	MÉDIA	MEDIANA
JRip	50,8	50,0	52,1	48,8	**58,8**	44,4	30,0	50,0	**28,6**	**45,9**	**50,0**
NNge	50,8	48,1	52,1	56,1	**61,8**	55,6	35,0	35,7	**28,6**	**47,1**	**50,8**
OneR	50,8	51,9	**54,2**	53,7	50,0	48,1	45,0	50,0	**42,9**	**49,6**	**50,0**
Prism	34,4	40,7	**29,2**	31,7	**50,0**	40,7	35,0	42,9	42,9	**38,6**	**40,7**
Ridor	47,5	55,6	47,9	61,0	50,0	48,1	45,0	50,0	**71,4**	**52,9**	**50,0**
ADTree	50,8	57,4	56,3	**58,5**	50,0	40,7	35,0	35,7	**28,6**	**45,9**	**50,0**
J48	50,8	48,1	**41,7**	48,8	44,1	**55,6**	50,0	50,0	42,9	**48,0**	**48,8**
RandomTree	55,7	57,4	54,2	53,7	55,9	59,3	40,0	**21,4**	**71,4**	**52,1**	**55,7**
REPTree	47,5	48,1	**52,1**	46,3	47,1	48,1	45,0	**35,7**	42,9	**45,9**	**47,1**
SimpleCart	47,5	**57,4**	52,1	53,7	50,0	48,1	50,0	57,1	**42,9**	**51,0**	**50,0**
SVM - kernel:linear	49,2	44,4	50,0	48,8	**61,8**	55,6	45,0	50,0	**0,0**	**45,0**	**49,2**
SVM - kernel:polynomial	47,5	**48,1**	47,9	46,3	47,1	48,1	45,0	**35,7**	42,9	**45,4**	**47,1**
SVM - kernel:RBF	47,5	**48,1**	47,9	46,3	47,1	48,1	45,0	**28,6**	42,9	**44,6**	**47,1**
SVM - kernel:sigmoid	47,5	**48,1**	47,9	46,3	47,1	48,1	45,0	**28,6**	42,9	**44,6**	**47,1**
Naive Bayes	50,8	50,0	50,0	53,7	52,9	51,9	50,0	**35,7**	**57,1**	**50,2**	**50,8**
Rede Neural	47,5	46,3	52,1	43,9	**52,9**	48,1	45,0	42,9	**0,0**	**42,1**	**46,3**
KNN	47,5	37,0	39,6	**36,6**	50,0	55,6	45,0	**57,1**	**57,1**	**47,3**	**47,5**
MÉDIA	**48,5**	**49,2**	**48,6**	**49,1**	***51,6***	**49,7**	**42,9**	**41,6**	**40,3**		
MEDIANA	**47,5**	**48,1**	***50,0***	**48,8**	***50,0***	**48,1**	**45,0**	**42,9**	**42,9**		
DESVIO PADRO	**4,26**	**5,63**	**6,59**	**7,41**	**5,21**	**5,25**	**5,88**	**10,49**	**19,71**		

Questão 3. Quão eficientes são os algoritmos de classificação caixa branca, por regras e por árvores de decisão, no que se refere, respectivamente, aos atributos nos antecedents das regras e ao nível de profundidade das árvores, observados particularmente no momento de início do curso?

Na execução dos algoritmos de classificação para o experimento 3, foi utilizada a técnica de Validação Cruzada, com 10 folds, conforme metodologia proposta baseada nos trabalhos de referência. Aqui o objetivo é responder à Questão de Pesquisa 3, que sugere uma análise das árvores com ou sem podas, assim como com podas gradativas. Foram utilizados aqui apenas os 5 algoritmos de classificação de árvore de decisão. Vale salientar que para cada

algoritmo de árvore, existem parâmetros distintos e diversos para o processo de poda da árvore, assim como para cada algoritmo a árvore se comporta de forma diferente quando podada. Porém, para todos os resultados apresentados, as árvores mantiveram nível de profundidade máximo de 1, 2, 3, 4, 5 e ilimitado.

Observa-se uma leve tendência geral de queda no desempenho geral, gradativamente com o aumento do tamanho das árvores. O Desvio Padrão também de mostrou menor para árvores maiores, utilizando-se todos os atributos disponíveis em T_0. O fato de que a média e mediana não apresentarem distâncias muito grandes leva a crer que não houve valores discrepantes de acurácia durante a análise gradativa com podas.

Na Tabela 10, verifica-se o nível de acurácia alcançado por cada algoritmo em cada situação de poda. Os valores em azul e vermelho indicam respectivamente os maiores e menores valores alcançados, em linha e em coluna. O algoritmo REPTree se mostrou uniforme em acurácia, para todos os limites de tamanho de árvore, o que mostra que ele não sofre influência significativa com as podas, além disso ele também manteve o maior nível médio de acurácia quando comparado às outras árvores. Além disso, em termos de eficiência, vale reconhecer que o ADTree e o RandomTree se mostraram melhores, entregando as maiores acurácias com árvores de menor profundidade, quando utilizando dados sem Seleção de Atributos, respondendo primeiramente a questão de Pesquisa 3.

Tabela 11 - Acurácia dos Modelos de Predição em Árvore de Decisão relativos à profundidade da árvore

	NP 1	NP 2	NP 3	NP 4	NP 5	Árvore Completa	MÉDIA	MEDIANA
ADTree	**58,8**	51,5	50,0	47,1	48,5	**45,6**	**50,2**	**49,3**
J48	**35,3**	38,2	38,2	35,3	33,8	**42,6**	**37,3**	**36,8**
RandomTree	**58,8**	55,9	50,0	45,6	47,1	**44,1**	**50,2**	**48,5**
REPTree	52,9	52,9	52,9	52,9	52,9	52,9	**52,9**	**52,9**
SimpleCart	**48,5**	**45,6**	47,1	**45,6**	**45,6**	**48,5**	**46,8**	**46,3**
MÉDIA	**50,9**	**48,8**	**47,6**	**45,3**	**45,6**	**46,8**		
MEDIANA	**52,9**	**51,5**	**50,0**	**45,6**	**47,1**	**45,6**		
DESVIO PADRÃO	**9,73**	**7,01**	**5,66**	**6,36**	**7,13**	**4,08**		

Os resultados apresentados após o processo de Seleção de Atributos mostraram um comportamento geral inverso ao apresentado quando não é feita a seleção.

Tabela 12- Acurácia dos Modelos de Predição em Árvore de Decisão relativos à profundidade da árvore com Seleção de Atributos

	NP 1	NP 2	NP 3	NP 4	NP 5	Árvore Completa	MÉDIA	MEDIANA
ADTree	47,1	50,0	**45,6**	51,5	52,9	**60,3**	**51,2**	**50,7**
J48	41,2	41,2	**39,7**	**39,7**	**39,7**	**48,5**	**41,7**	**40,4**
RandomTree	**44,1**	**44,1**	**54,4**	51,5	**44,1**	47,1	**47,5**	**45,6**
REPTree	52,9	52,9	52,9	52,9	52,9	52,9	**52,9**	**52,9**
SimpleCart	**45,6**	50,0	50,0	**52,9**	47,1	47,1	**48,8**	**48,5**
MÉDIA	**46,2**	**47,6**	**48,5**	**49,7**	**47,3**	**51,2**		
MEDIANA	**45,6**	**50,0**	**50,0**	**51,5**	**47,1**	**48,5**		
DESVIO PADRÃO	**4,36**	**4,84**	**5,98**	**5,64**	**5,74**	**5,64**		

Na tabela 12 confirma-se tal tendência a partir da observação da concentração dos resultados de maior acurácia nas árvores mais profundas, Tamanho 3 ou maior. Por sua, vez as piores acurácias se encontram concentradas nas árvores menores. Pode-se concluir como fator relevante para este comportamento o fato de que dados mais “ruidosos”, em outras palavras, que possuem atributos que não são relevantes para a aprovação do aluno, tendem a contribuir negativamente para os processos de aprendizagem do algoritmo, direcionando à criação de modelos de árvore “poluídos”, que utilizam estes atributos pouco relevantes como nós decisores da árvore. Porém, uma das características relevantes de uma Àrvore de Decisão é reduzir o impacto dos atributos menos relevantes. Quanto mais próximo da raiz estiver um atributo irrelevante, maior será o impacto negativo na acurácia obtida para a predição. Como resposta adicional à Questão 3, pode-se considerar este comportamento apresentado, acrescentando que a eficiência das árvores será afetada pelo processo de poda de acordo com a qualidade dos atributos de entrada.

Mais uma vez observa-se que a seleção de atributos elevou as acurácias inferiores: J48 melhorou de 37,3% para 41,7%. Observa-se também que, utilizando-se apenas atributos mais relevantes, o algoritmo REPTree se mostra desta vez mais eficiente: além de entregar a maior acurácia com árvore de

profundidade 1, não sofre alteração de acurácia com as podas.

Uma conclusão essencial que pode ser obtida com este experimento é que os níveis de acurácia dos algoritmos de classificação, quando submetidos a atributos dos alunos disponíveis para o professor no momento chamado T_0, ou momento inicial do curso, não são satisfatórios, o que leva a crer que uma predição realizada no momento T_0 apenas com os dados socioeconômicos não é segura o suficiente para dar suporte ao professor, sistema ou especialista em EDM. Acredita-se então, que uma amostra maior de dados para treino e teste permita uma predição mais acurada em T_0.

Questão 4. A partir de qual período de tempo no curso, consegue-se obter níveis de desempenho satisfatórios na tarefa predição antecipada de insucesso, bem como qual o comportamento dos algoritmos nos períodos de tempo seguintes, após tal momento cedo?

O Experimento 4 destina-se a responder a Questão de pesquisa 4, considerando-se uma análise temporal da eficiência dos 14 algoritmos no decorrer do curso, buscando encontrar o momento mais breve possível onde uma predição satisfatória pode ser realizada. Para este experimento exclusivamente, foram mantidos os dados das notas das atividades dos alunos no Huxley.

Analisando-se a Tabela 9, constata-se o que já era, de algum modo, esperado: com a inserção dos dados sobre as notas dos alunos nas atividades, unindo-os aos dados iniciais dos alunos disponíveis em T_0, independente do algoritmo de classificação utilizado, sempre existe um incremento na acurácia, proporcional ao tempo de curso e consequentemente proporcional ao número de atividades realizadas. Percebe-se também uma constância no desvio padrão ao longo de todo o período, indicando variabilidade constante dos resultados. Nota-se ainda que após a tempo T4, o nível geral de acurácia na predição de todos os algoritmos se mantenha praticamente constante ou com pequena variação até o final do curso.

Com respeito à eficiência dos algoritmos de classificação nas categorias Caixa Branca: Regra, Caixa Branca: Árvore de Decisão e Caixa preta, em grupos separados, todos eles apresentam crescimento significativo na acurácia

até T4 e pouca variação a partir de então. Há indícios, portanto, de que a predição apresenta resultados promissores a partir de 50% do curso, para alguns algoritmos.

Analisando-se os valores individuais de acurácia de cada atributo, apresentados na Tabela 12, percebe-se uma tendência geral de crescimento. A máxima acurácia apresentada foi do algoritmo Ridor, no instante T8, com 91,2%.

Tabela 12 - Acurácias da predição dos algoritmos de classificação relativos ao tempo de curso e informações adquiridas nas atividades de aprendizagem.

Algoritmo	T0	T1	T2	T3	T4	T5	T6	T7	T8	MÉDIA	MEDIANA
JRip	55,9	57,4	64,7	75,0	91,2	89,7	85,3	91,2	85,3	77,3	85,3
NNge	54,4	55,9	60,3	80,9	82,4	82,4	82,4	83,8	88,2	74,5	82,4
OneR	63,2	51,5	75,0	82,4	86,8	86,8	86,8	86,8	89,7	78,8	86,8
Prism	19,1	26,5	27,9	29,4	39,7	44,1	48,5	47,1	55,9	37,6	39,7
Ridor	52,9	58,8	70,6	76,5	89,7	88,2	88,2	89,7	91,2	78,4	88,2
ADTree	51,5	57,4	67,6	82,4	91,2	85,3	85,3	88,2	86,8	77,3	85,3
J48	35,3	42,6	64,7	76,5	89,7	91,2	91,2	91,2	83,8	74,0	83,8
RandomTree	44,1	54,4	55,9	66,2	67,6	72,1	79,4	73,5	64,7	64,2	66,2
REPTree	52,9	44,1	55,9	73,5	88,2	89,7	89,7	89,7	86,8	74,5	86,8
SimpleCart	48,5	52,9	70,6	82,4	92,6	92,6	92,6	92,6	89,7	79,4	89,7
SVM - kernel:linear	55,9	60,3	64,7	70,6	85,3	82,4	82,4	85,3	88,2	75,0	82,4
SVM - kernel:polynomial	54,4	60,3	60,3	69,1	83,8	82,4	80,9	83,8	86,8	73,5	80,9
SVM - kernel:RBF	58,8	66,2	69,1	73,5	80,9	83,8	83,8	85,3	89,7	76,8	80,9
SVM - kernel:sigmoid	45,6	45,6	45,6	44,1	45,6	47,1	47,1	47,1	48,5	50,3	45,6
Naive Bayes	47,1	51,5	58,8	73,5	83,8	85,3	86,8	85,3	88,2	73,4	83,8
Rede Neural	39,7	50,0	54,4	57,4	72,1	76,5	76,5	76,5	85,3	65,4	72,1
KNN	54,4	55,9	54,4	58,8	61,8	61,8	64,7	64,7	72,1	60,9	61,8
MÉDIA	**49,0**	**52,4**	**60,0**	**68,9**	**78,4**	**78,9**	**81,7**	**80,1**	**81,2**		
MEDIANA	**52,9**	**54,4**	**60,3**	**73,5**	**83,8**	**83,8**	**83,8**	**85,3**	**86,8**		
DESVIO PADRÃO	**10,39**	**9,09**	**11,15**	**14,41**	**15,98**	**14,65**	**10,68**	**14,33**	**12,90**		

Praticamente todos os algoritmos apresentaram como seus piores resultados de acurácia o desempenho em T0, ou seja, no início do curso, quando não se tem nenhuma avaliação de conhecimento do aluno. Já as melhores acurácias individuais de cada algoritmo se encontram, em sua maioria, a partir de T4.

Observa-se como tendência geral que, mesmo quando aplicada a seleção de atributos, também em T4 a média entre as acurácias dos algoritmos de

classificação chega próximo de 80% e continua até T8 com pouco crescimento, basicamente flutuando entre 80% e 90% de média. Podemos perceber também comparando as tabelas 12 e 13 que a seleção de atributos elevou a acurácia mínima de 19,1% para 26,5% e a média das acurácias obtidas em T8 de 81,2% para 82,9%.

Na análise dos resultados por tipo de algoritmo, percebe-se que os algoritmos caixa branca de árvore de decisão se destacaram em eficiência temporal, alcançando uma média de acurácia bem próximo a 90% em T4. O Algoritmo SimpleCart se destacou dentre todos, chegando a exatos 92,6% de T4 a T7.

Na Tabela 13 é possível confirmar os algoritmos mais eficientes, que neste caso foram os de árvore de decisão, tendo obtido suas melhores acurácias já em T4. JRip também se destacou nesta situação, apresentado 91,2 em T4, superando os demais algoritmos de regras.

Tabela 13 - Acurácias da predição dos algoritmos de classificação relativos ao tempo de curso e informações adquiridas nas atividades de aprendizagem com Seleção de Atributos.

Algoritmo	T0	T1	T2	T3	T4	T5	T6	T7	T8	MÉDIA	MEDIANA
JRip	45,6	69,1	70,6	80,9	91,2	88,2	91,2	85,3	86,8	78,8	85,3
NNge	54,4	58,8	64,7	75,0	82,4	80,9	80,9	82,4	88,2	74,2	80,9
OneR	48,5	63,2	75,0	82,4	86,8	86,8	86,7	86,8	89,7	78,4	86,7
Prism	26,5	35,3	36,8	50,0	47,1	54,4	57,4	55,9	55,9	46,6	50,0
Ridor	54,4	58,8	75,0	77,9	88,2	89,7	89,7	89,7	89,7	79,2	88,2
ADTree	48,5	61,8	70,6	80,9	86,8	88,2	86,8	85,3	88,2	77,4	85,3
J48	42,6	41,2	70,6	77,9	88,2	85,3	85,3	85,3	83,8	73,4	83,8
RandomTree	51,5	45,6	63,2	61,8	77,9	77,9	77,9	72,1	77,9	67,3	72,1
REPTree	50,0	44,1	55,9	72,1	88,2	88,2	88,2	88,2	86,8	73,5	86,8
SimpleCart	42,6	55,9	70,6	82,4	92,6	92,6	92,6	92,6	89,7	79,1	89,7
SVM - kernel:linear	45,6	55,9	58,8	67,6	82,4	86,8	86,8	86,8	91,2	73,5	82,4
SVM - kernel:polynomial	45,6	44,1	45,6	50,0	48,5	50,0	50,0	50,0	50,0	48,2	50,0
SVM - kernel:RBF	47,1	64,7	70,6	79,4	85,3	86,8	86,8	85,3	88,2	77,1	85,3
SVM - kernel:sigmoid	47,1	44,1	72,1	79,4	82,4	85,3	88,2	82,4	88,2	74,3	82,4
Naive Bayes	51,5	63,2	66,2	76,5	86,8	88,2	88,2	86,8	89,7	77,4	86,8
Rede Neural	50,0	52,9	55,9	61,8	72,1	75,0	76,5	75,0	85,3	67,2	72,1
KNN	57,4	61,8	64,7	67,6	69,1	72,1	73,5	73,5	79,4	68,8	69,1
MÉDIA	**47,6**	**54,1**	**63,9**	**72,0**	**79,8**	**81,0**	**81,6**	**80,2**	**82,9**		
MEDIANA	**48,5**	**55,9**	**66,2**	**76,5**	**85,3**	**86,8**	**86,8**	**85,3**	**88,2**		

DESVIO PADRÃO	**6,80**	**9,90**	**10,52**	**10,60**	**13,53**	**12,12**	**11,79**	**11,71**	**11,87**

Ainda no experimento 4, e referente à questão de pesquisa 4, indo mais além no que diz respeito à qualidade dos dados de entrada, buscou-se uma análise temporal mais ousada e ao mesmo tempo experimental, utilizando-se como dados de entrada em cada marco de tempo de T0 a T8 apenas os dados das atividades realizadas no Huxley, ou seja, foram deixados de lado todos os dados socioeconômicos a partir de T0. Uma abordagem semelhante foi feita por (MARQUEZ-VERA et al ,2016) porém com o objetivo de predizer evasão de alunos. Em T0, para os resultados que serão mostrados a seguir, foram utilizados os dados socioeconômicos após seleção de atributos, a fim de permitir um comparativo entre T0 com atributos socioeconômicos mais relevantes e T1 em diante sem dados socioeconômicos e apenas com informações obtidas através das atividades realizadas.

Os algoritmos de Árvore de Decisão assim como os de Caixa Preta, apresentaram um desvio padrão inferior em termos de médias dos resultados das acurácias quando comparados aos algoritmos de Regras, sinalizando pouca variação entre as acurácias,

As melhores acurácias mais uma vez foram alcançadas pelos algoritmos de Caixa Preta. Embora alguns resultados máximos tenham sido alcançados apenas em T4, observa-se desempenhos bastante altos em acurácia já no momento T3: a exemplo do algoritmo SVM com kernel = sigmoid, que chegou a 86,8% de acurácia.

Tal melhoria de desempenho pode ser direcionada a 2 fatores correlacionados: 1) O ruído gerado pelos atributos socioeconômicos nas predições temporais e 2) O fato de que as atividades de aprendizagem são um elemento valioso para a tarefa de predição, podendo por si só permitir a extração de informações relevantes para a tomada de decisão dos educadores.

Pode-se traduzir em outras palavras que a cada 10 alunos, 8 o professor pode ter uma predição assertiva em 45 dias de curso, ainda antes da primeira avaliação bimestral, e considerando apenas a aplicação de 3 atividades. Em sendo possível tais atividades com intervalos de tempo menores entre elas, ou

seja, atividades aplicadas semanalmente, pode-se acreditar a partir desta análise que é possível alcançar predições ainda mais eficientes com menos tempo.

Tabela 14 - Acurácias da predição dos algoritmos de classificação relativos ao tempo de curso e informações adquiridas nas atividades de aprendizagem com Seleção de Atributos utilizando apenas atributos das atividades do Huxley.

Algoritmo	T0	T1	T2	T3	T4	T5	T6	T7	T8	MÉDIA	MEDIANA
JRip	45,6	66,2	73,5	85,3	89,7	88,2	88,2	86,8	89,7	79,2	86,8
NNge	54,4	54,4	63,2	69,1	85,3	89,7	89,7	88,2	88,2	75,8	85,3
OneR	48,5	63,2	75,0	82,4	86,8	86,8	86,8	86,8	89,7	78,4	86,8
Prism	26,5	48,5	54,4	58,8	44,1	55,9	61,8	60,3	58,8	52,1	55,9
Ridor	54,4	66,2	72,1	76,5	91,2	88,2	88,2	88,2	89,7	79,4	88,2
ADTree	48,5	67,6	69,1	82,4	89,7	85,3	88,2	88,2	91,2	78,9	85,3
J48	42,6	61,8	72,1	82,4	89,7	85,3	85,3	85,3	85,3	76,6	85,3
RandomTree	51,5	70,6	69,1	82,4	83,8	86,8	86,8	89,7	89,7	78,9	83,8
REPTree	50,0	60,3	70,6	77,9	88,2	88,2	88,2	88,2	88,2	77,8	88,2
SimpleCart	42,6	70,6	73,5	82,4	92,6	85,3	85,3	86,8	89,7	78,8	85,3
SVM - kernel:linear	45,6	67,6	77,9	83,8	92,6	91,2	91,2	91,2	95,6	81,9	91,2
SVM - kernel:polynomial	45,6	69,1	67,6	83,8	86,8	82,4	83,8	83,8	86,8	76,6	83,8
SVM - kernel:RBF	47,1	66,2	75,0	80,9	92,6	91,2	91,2	86,8	89,7	80,1	86,8
SVM - kernel:sigmoid	47,1	70,6	77,9	86,8	92,6	89,7	91,2	86,8	94,1	81,9	86,8
Naive Bayes	51,5	66,2	72,1	80,9	88,2	89,7	88,2	86,8	92,6	79,6	86,8
Rede Neural	50,0	64,7	76,5	85,3	91,2	92,6	92,6	91,2	89,7	81,5	89,7
KNN	57,4	70,6	70,6	80,9	85,3	88,2	88,2	88,2	88,2	79,7	85,3
MÉDIA	47,6	65,0	71,2	80,1	86,5	86,2	86,8	86,1	88,1		
MEDIANA	48,5	66,2	72,1	82,4	89,7	88,2	88,2	86,8	89,7		
DESVIO PADRÃO	6,80	6,02	5,74	6,82	11,29	8,22	6,86	6,90	7,93		

Como resposta à Questão de pesquisa 4, pode-se concluir a partir dos resultados mostrados no experimento, que o momento T0, em outra palavras, a partir de aproximadamente 3 atividades de aprendizagem, independendo do algoritmo de classificação, já é possível obter uma acurácia significativa, nos melhores casos acima de 85%, na predição dos alunos que se encontram em risco de serem reprovados na disciplina, Isso chama atenção de que a parcela de informação agregada com a realização de atividades rotineiras durante o curso se mostra bastante relevante e auxilia diretamente na predição por parte dos algoritmos de classificação, permitindo inclusive a criação de modelos capazes de apoiar o professor na identificação dos alunos e atuação preventiva.

Questão 5: Como prover formas de explicação que possam permitir ao professor compreender adequadamente as informações produzidas no processo de predição aplicado, considerando os algoritmos caixa branca?

De acordo com os experimentos realizados e descritos anteriormente, os modelos caixa branca apresentaram desempenhos satisfatórios, tanto na etapa antes de iniciar o curso, quanto na predição antecipada e períodos de tempo seguintes. Assim, no momento inicial a escolha foi a de um algoritmo de árvore de decisão, no caso o J48, que é uma implementação do C4.5, sendo uma árvore muito simples e de explicabilidade fácil, pois se baseou para predição satisfatória apenas no atributo nota do ENEM relativa à Matemática. Relativamente à predição antecipada e momentos posteriores, o algoritmo caixa branca com resultado mais satisfatório foi o JRip, implementação no WEKA do Ripper, sendo um algoritmo que produz regras de classificação. Nesse caso também as regras se mostraram de fácil entendimento, sendo compreensíveis quanto a obter informação de explicação delas.

5.4 Estudo Comparativo

Dando continuidade ao estudo realizado no experimento 4, buscando uma maior exploração à resposta da Questão de Pesquisa 4, pode-se concluir, a partir de todos os resultados obtidos no experimento 4, que com a utilização de atividades de fixação e avaliativas quinzenais, um período de tempo que ficou bem marcado como tendo uma predição a níveis satisfatórios é o de 45 dias, ou seja, entre T0 e T3. Fundamenta-se tal conclusão a partir da exploração média considerada de todos os algoritmos, onde comparando-se os de Regras, Árvores de Decisão e Caixa preta, todos apresentaram uma mesma tendência, com leves variações. Uma outra conclusão obtida de forma paralela foi a relevância dos atributos das atividades aplicadas com os alunos, que se mostraram significativamente mais valiosos do que os socioeconômicos. Mesmo quando utilizados de forma conjunta com as notas das atividades do Huxley, os atributos socioeconômicos não agregaram informação relevante ao treino e teste dos algoritmos, de forma geral, desfavorecendo a eficiência alcançada pelos algoritmos quando utilizados tendo apenas dados das atividades como entrada.

Considerando que os atributos socioeconômicos não representam dados que agreguem informação adicional ao processo de predição quando utilizando

atividades de aprendizagem, um estudo mais representativo, relacionado também à questão de pesquisa 4, foi realizado utilizando um conjunto maior de instâncias, buscando dessa forma aproximar-se do quantitativo de instâncias utilizados por (Marquez-Vera et al., 2013; Marquez-Vera et al., 2016). Apenas dados do Huxley e 5 socioeconômicos contidos no Sistema Acadêmico foram utilizados como atributos de entrada, porém referente ao período de 2013.2 a 2017.1, das mesmas disciplinas, Engenharia da Computação e Ciência da Computação. No total desta experimentação, foram utilizados apenas 14 atributos, sendo 6 de atividades, 5 socioeconômicos extraídos do sistema acadêmico da Ufal SIEWEB, e 3 atributos acadêmicos: Notas das Avaliações Bimestrais 1 e 2 e Conceito Final. No total foram 544 instâncias de entrada, considerando os alunos matriculados nas 2 disciplinas no período mencionado, de 4 anos. A partir de T1, os 5 atributos socioeconômicos foram desconsiderados, passando a ser considerado apenas as notas das atividades do Huxley de T1 em diante.

Tabela 15 - Acurácias da predição dos algoritmos de classificação relativos ao tempo de curso e informações adquiridas nas atividades de aprendizagem utilizando apenas atributos das atividades do Huxley e referente aos períodos de 2013.2 a 2017.1

Algoritmo	T0	T1	T2	T3	T4	T5	T6	T7	T8	MÉDIA	MEDIANA
JRip	59,1	71,0	73,9	75,1	83,4	84,4	84,5	83,6	92,5	78,6	83,4
NNge	55,9	61,9	68,2	72,5	79,2	82,7	85,0	85,3	92,3	75,9	79,2
OneR	56,0	68,4	75,2	75,2	83,6	83,6	83,6	82,6	91,0	77,7	82,6
Prism	44,0	46,9	46,4	49,2	47,6	52,0	53,1	58,0	62,7	51,1	49,2
Ridor	58,3	68,1	72,1	73,8	78,2	82,1	82,9	82,9	91,7	76,7	78,2
ADTree	58,8	70,5	73,6	75,4	82,7	82,6	82,7	85,5	91,5	78,2	82,6
J48	57,7	70,5	72,6	76,5	83,9	81,8	82,2	82,9	90,9	77,7	81,8
RandomTree	55,9	66,1	64,8	65,5	77,0	82,7	81,1	82,6	92,4	74,2	77,0
REPTree	59,6	69,7	71,7	75,6	82,6	83,2	83,9	83,9	91,2	77,9	82,6
SimpleCart	58,8	70,0	71,8	75,2	84,2	83,7	84,0	83,1	92,0	78,1	83,1
SVM - kernel:linear	58,0	69,2	73,5	73,1	83,6	84,2	84,7	85,3	92,0	78,2	83,6
SVM - kernel:polynomial	55,5	68,2	73,3	74,8	81,6	83,1	83,1	83,6	91,2	77,1	81,6
SVM - kernel:RBF	58,3	69,1	73,9	74,3	83,4	84,5	84,9	85,7	92,8	78,5	83,4
SVM - kernel:sigmoid	49,5	49,5	49,5	49,5	71,7	74,9	75,2	76,1	83,1	64,3	71,7
Naive Bayes	57,3	66,8	71,2	71,5	79,6	81,4	81,9	82,7	89,3	75,7	79,6
Rede Neural	58,5	68,1	73,1	72,1	82,4	84,4	84,4	84,7	92,0	77,7	82,4
KNN	55,9	63,0	65,0	68,7	80,1	83,4	85,0	84,5	90,7	75,1	80,1
MÉDIA	56,3	65,7	68,8	70,5	79,1	80,9	81,3	81,9	89,4		
MEDIANA	57,7	68,2	72,1	73,8	82,4	83,1	83,6	83,6	91,5		
DESVIO PADRÃO	3,94	7,06	8,41	8,41	8,75	7,77	7,63	6,55	7,22		

A Tabela 15, enfatiza a tendência geral apresentada anteriormente e confirma o resultado obtido para 2013.1. As acurácias alcançadas ficaram bem próximas numericamente, devido à relação de número de instâncias alto para poucos atributos, o que tende a resultados mais estáveis e com menos variações no decorrer do curso. Curiosamente, um fato que chamou atenção é o nível da acurácia alcançado por (Marquez-Vera et al. ,2013) com o algoritmo Prism: no referido trabalho ele alcança a marca de 94,7% de acurácia, enquanto a maior acurácia alcançada por este algoritmo neste estudo realizado é de 65,7%.

O fato que mais aproxima e ao mesmo tempo reforça os resultados encontrados em (Marquez-Vera et al , 2016) é a conclusão permitida a partir da Questão de Pesquisa 4 deste estudo, que indica o T3 (eventualmente T2) é o momento mais favorável para a realização de uma predição antecipada. (Marquez-Vera et al., 2016) identificou e recomendou em seu trabalho anterior que predições confiáveis podem ser alcançadas dentro da faixa de 4 a 6 semanas de curso, com a aplicação de atividades semanais para o ganho de informação sobre os alunos. Portanto, o resultado aqui obtido se mostra mais interessante do que o equivalente encontrado em (Marquez-Vera et all. ,2016).

5.5 Síntese e análise dos resultados

A abordagem proposta se mostrou adequada em responder satisfatoriamente a predição inicial, antes de começar o curso, assim se utilizando de dados trazidos pelos estudantes, assim como na predição antecipada durante o curso, continuando com boas predições em períodos de tempo posteriores. Assim, verificou-se um resultado preditivo de boa qualidade para o momento inicial obtido com um algoritmo Caixa branca, baseado em árvore, destacando-se o RepTree, mas obtendo desempenho próximo com mais dois algoritmos baseados em árvore, J48 e ADTree, operando apenas sobre o atributo associado à nota de Matemática no ENEM, salientando-se que este algoritmos apresentaram uma acurácia similar ao SVM, todos se mostraram adequados em termos de compreensibilidade dos modelos.

Portanto, esse desempenho pode ser comparado ao desempenho no momento inicial do curso obtido no estudo em (Marquez-Vera at al.,2016), sendo que neste havia mais atributos envolvidos na predição.

Quanto ao desempenho aferido para o durante o curso, destaca-se o resultado satisfatório da predição antecipada que pode ser obtido já entre T2 e T3, portanto com aproximadamente 30 dias do início do curso semestral. Neste caso, teve o destaque para um algoritmo baseado em regras de classificação, no caso o JRip, portanto um algoritmo caixa branca, assim oferecendo facilidade para compreensibilidade, sendo que novamente a predição satisfatória foi obtida com apenas poucos atributos acadêmicos, no caso notas no sistema TheHuxley, tendo outra vez um desempenho comparável ao SVM. Quanto à comparação com os resultados em (Marquez-Vera et al., 2016), obteve-se resultado superior, pois neste trabalho se utilizou de muitos atributos, inclusive sócioeconômicos, tendo o resultado satisfatório antecipado entre as semanas 4 e 6. Na abordagem proposta, perseguiu-se o objetivo de economicidade sobre os dados e atributos utilizados, o que se mostrou menos custoso que o que se apresentou nesse trabalho correlato.

Dados os estudos experimentais realizados, pode-se consolidar uma solução para abordagem proposta como sendo formada por dois tipos de algoritmo caixa branca: um algoritmo baseado em árvore para predição no momento inicial, no caso poderia ser o J48, e um baseado em regras para a predição antecipada e períodos de tempo posteriores, no caso usando o JRip. Com essa solução, a compreensibilidade ou explicação dos modelos fica facilitada, notadamente por ser uma característica potencial dos algoritmos caixa branca.

Numa tentativa de verificar o potencial de generalização da solução obtida, verificando a possibilidade de aplicar a abordagem em outras situações, foi realizado um experimento em uma turma de programação introdutória no nível técnico, observando uma adequação desta abordagem, pois produziu um comportamento bem similar. No entanto, isso representa apenas um indício, não garantindo a portabilidade da proposta no sentido da generalização.

A discussão anteriormente apresentada ressaltou a qualidade dos resultados dos estudos experimentais, relativamente aos trabalhos

posicionados como mais próximos, dando indicativos favoráveis. No entanto, na perspectiva de análise dos resultados obtidos com vistas à utilidade da informação produzida e, portanto, ao potencial de uso prático pelos professores em suas atividades de tomada de decisão pedagógica, cabe aqui apresentar algumas reflexões.

Com respeito ao resultado obtido para o tempo inicial (T_0), usando-se dados oriundos da fonte ENEM, particularmente, indicando apenas a nota de Matemática como atributo selecionado para predição, tem-se algo pertinente e esperado. No entanto, pode-se questionar a suficiência da informação obtida a partir deste parâmetro, pois fatores como a temporalidade e outras circunstâncias intrínsecas ou mesmo extrínsecas ao estudante, podem influenciar no desempenho de um tal estudante na ocasião do ENEM, sendo uma ameaça à validade dos experimentos aqui realizados. Assim, numa tentativa de buscar mais confiança na informação a ser obtida para cada estudante, sugere-se um diagnóstico imediatamente antes do início do curso, pois isso produzirá uma informação que refletirá melhor a situação cognitiva atual do estudante. Ademais, há que se considerar atributos de outra natureza, por exemplo, os de natureza afetiva ou mesmo social. Além disso, pode-se refletir melhor sobre uma questão de base, qual seja: o valor da classificação em si adotada, atribuída ao estudante como tendendo ao insucesso ou ao sucesso. Neste caso, há que se considerar experimentações e discussões minuciosas com os professores de programação, verificando-se aspectos de utilidade e de adequação.

Quanto aos resultados de predição obtidos no tempo o mais cedo possível, assume-se como satisfatórios, sendo atingidos a partir de 30 dias após o início do curso, trazendo um valor na qualidade da informação gerada para, em alguma etapa, chegar ao professor, deste modo, tendo potencial para ajudá-lo, notadamente no monitoramento da aprendizagem do estudante e nas decisões sobre eventuais mudanças nos planos de ensino, sempre que apropriado. No entanto, apesar de bons resultados técnicos obtidos, de um ponto de vista mais prático e útil para auxiliar mais efetivamente o professor, os resultados certamente seriam incrementados com a adição de atributos com dados mais diretamente relacionados ao processo de resolução de problemas.

Tal como o que foi observado para predição no momento antes do início do curso, ressalta-se aqui também a importância de buscar um *feedback* dos professores quanto à questão do tipo de classificação considerado, tendo em consideração aspectos de utilidade e adequação.

6 CONSIDERAÇÕES FINAIS

Neste documento se apresentou os aspectos principais relacionados à concepção, ao desenvolvimento e à avaliação de uma abordagem preditiva voltada para identificar, com antecedência, envolvendo o momento inicial e o decorrer do curso, estudantes com propensão ao insucesso em disciplinas de programação introdutória.

A pesquisa apresentada nesta tese faz parte da área de Inteligência Artificial em Educação, em seu propósito de usar técnicas para desenvolver soluções para melhorar a efetividade do processo educacional. Em particular, esta pesquisa usa técnicas de mineração de dados educacionais, visando contribuir para avançar o estado da tecnologia em favor da educação online, pretensamente provendo o professor com informações, sobre seus alunos, com alguns requisitos de qualidade, notadamente as que chegam em momentos oportunos, sejam confiáveis e compreensíveis. Neste sentido, a abordagem proposta teve a pretensão de auxiliar no gerenciamento apropriado do curso online, podendo resultar em suporte efetivo aos estudantes. Especificamente, a pesquisa se mostra relevante como possibilidade de agregar tecnologia aos atuais ambientes virtuais de aprendizagem, disponibilizando serviço de qualidade voltado para predição de desempenho acadêmico de estudantes de programação. Assim, focalizou-se em monitorar e oferecer diagnóstico preditivo de uma determinada turma, considerando-se diferentes períodos de tempo ao longo do curso, inclusive o momento inicial do curso, permitindo ao professor aproveitar cada informação gerada para tomar decisões pedagógicas, além do que serviria para incrementar soluções sobre modelo de estudante, o que também permitiria agentes de software considerar tal informação em suas decisões sobre o que recomendar recursos pedagógicos.

Na perspectiva acima, especificamente, realizou-se nesta pesquisa alguns estudos avaliativos sobre modelos preditivos. Neste sentido, investigou-se minuciosamente quais seriam as técnicas de predição mais eficazes na identificação dos estudantes propensos ao insucesso, através de um estudo comparativo sobre o potencial e eficácia das quatro técnicas (árvore de decisão, máquina de vetores de suporte, rede neural e naive bayes)

analisadas. Avaliou-se a eficácia das técnicas de predição em duas fontes de dados diferentes e independentes, uma na modalidade de ensino presencial e a outra na modalidade de ensino a distância sobre as disciplinas de programação introdutória. Em seguida, ampliou-se o estudo para outras fontes de dados e investiu-se em algoritmos do tipo caixa branca, alguns baseados em árvore e outros em regras, mas tendo os resultados de algoritmos caixa preta como referência, desta vez com o intuito de oferecer compreensibilidade nos modelos utilizados, ainda tendo um compromisso com a boa acurácia preditiva. Finalmente, com as lições aprendidas nos estudos descritos no Capítulo 4 e suas complementações no Capítulo 5, consolidou-se, satisfatoriamente, a abordagem pretendida.

Numa retomada aos problemas de pesquisa norteadores desta tese, associados aos objetivos propostos, pode-se resumir o quanto eles foram atendidos do seguinte modo. Além do estudo de viabilidade sobre técnicas de mineração de dados para predição antecipada e com acurácia satisfatória e explicabilidade dos modelos, via modelos caixa branca, explorou-se nesta tese, com vistas à melhoria na eficiência, técnicas de seleção de atributos e ajustes nos algoritmos. Neste sentido, introduziu-se um processo para realizar modelagem preditiva no tempo, incrementalmente, desde a predição no marco zero do curso ao momento identificação, o mais cedo possível no curso, de estudantes com propensão insucesso, portanto realizando o objetivo da antecipada e etapas posteriores no curso, tudo isso numa solução de compromisso entre acurácia satisfatória e compreensibilidade do modelo.

Diante dos primeiros estudos realizados, descritos no Capítulo 4, as técnicas analisadas mostraram-se eficazes na identificação cedo dos estudantes propensos ao insucesso, inclusive os algoritmos do tipo caixa branca, onde os que tiveram os melhores desempenhos, ficaram em níveis próximos aos melhores caixa preta, notadamente o SVM. Após a realização das etapas de pré-processamento e ajustes finos todos algoritmos tiveram melhoras significativas em seus resultados. Ao fim do processo, nos dois primeiros estudos avaliativos, o algoritmo máquina de vetor de suporte proporcionou os melhores resultados tanto na modalidade de ensino presencial quanto na modalidade a distância. Na modalidade a distância, o algoritmo

alcançou uma taxa de f-measure de 92% com pelo menos 60% da disciplina realizada. Na modalidade presencial, o algoritmo alcançou uma taxa de f-measure de 83% com pelo menos 25% da disciplina realizada. Em seguida, no investimento em algoritmos caixa branca, obtendo-se acurácia acima de 90%.

Como uma informação valiosa obtida e confirmada com esta pesquisa em sua etapa conclusiva, conforme Capítulo 5, ressalta-se a importância do ganho em desempenho apresentado pela inserção de novos dados, mais especificamente os dados das avaliações regulares dos alunos, ficando evidente o crescimento do ganho de desempenho praticamente unânime de todos os algoritmos. T3 é o momento buscado pela terceira questão de pesquisa, considerando atividades quinzenais, ou seja, imediatamente antes da avaliação bimestral 1, porém em T2 já é possível realizar predições com acurácia acima de 82% quando considerando o uso de um algoritmo caixa branca. Considerando ainda que apenas com as notas das atividades de aprendizagem, desconsiderando dados socioeconômicos e seleção de atributos, é possível um nível de eficiência ainda superior na predição. Períodos interatividades menores também tendem à melhoria da predição antecipada.

A relevância de um trabalho pode ser avaliada também pelas oportunidades de trabalhos futuros que o mesmo provê. Então, buscando obter tal importância, são propostos alguns direcionamentos para novas pesquisas que podem ser identificados a partir desse estudo.

Outra questão que poderá ser investigada é o potencial de atingir melhores resultados usando novos fatores como: aspectos afetivos e motivacionais dos estudantes, além de variáveis ambientais, tais como laboratórios, metodologias pedagógicas.

REFERÊNCIAS BIBLIOGRÁFICAS

AGAPITO, J.B., Sosnovsky, S., Ortigosa, A. Detecting symptoms of low performance using production rules. Educational Data Mining, 2009

AGUDO-PEREGRINA A. F., Iglesias-Pradas S., Conde-González, M. A., and Hernández-García, A., “Can we predict success from log data in VLEs? Classification of interactions for learning analytics and their relation with performance in VLE-supported F2F and online learning,” Computers in Human Behavior, vol. 31, no. 1, pp. 542–550, 2014.

AGUIAR, J. J. B.; Fechine, J.M.; Costa, E.B. “Recomendação de Objetos de Aprendizagem baseada na Popularidade dos Objetos e nos Estilos de Aprendizagem dos Alunos.” Anais do 26o Simpósio Brasileiro de Informática na Educação (SBIE 2015), p. 1147-1156, 2015.

AHMAD F., Ismail N. H., and Aziz A. A., “The prediction of students’ academic performance using classification data mining techniques,” Appl. Math. Sci., vol. 9, no. 129, pp. 6415–6426, 2015.

AIRES, L. E-learning, educação online e educação aberta: contributos para uma reflexão teórica. Revista Iberoamericana de educación a distancia, v. 19:1, 2016.

ANDERSON, T. Toward a theory for on-line learning. In: ANDERSON, T. e ELLOUMI, F. (Ed.). Theory and practice of on-line learning. Athabasca, AB Athabasca University, p.33-60, 2004.

_____. Towards a theory of on-line learning. Theory and practice of on-line learning,v. 2, p. 15-44, 2008.

ANTUNES, Cláudia, “Anticipating student’s failure as soon as possible,” in Handbook of Educational Data Mining, C. Romero, S. Ventura, M. Pechenizkiy, and R. S. Baker, Eds. Boca Raton, FL: CRC Press Taylor, ch. 25, pp. 353–363,2011.

ARNOLD, K.E.; Pistilli, M.D. Course Signals at Purdue: Using Learning Analytics to Increase Student Success. Proceedings of the 2nd International

Conference on learning Analytics and Knowledge. ACM Press, p. 267–270, 2012.

ASIF, R.; Merceron, A.; Ali, S. A.; Haider, N. G. “Analyzing undergraduate students’ performance using educational data mining.” Computers & Education, vol. 113, pp. 177-194, 2017.

BARBER, R.; Sharkey, M. “Course Correction: Using Analytics to Predict Course Success.” Proceedings of the 2nd International Conference on Learning Analytics and Knowledge. ACM Press, 259-262, 2012.

BAKER, R.S.J.d., Yacef, K.. The State of Educational Data Mining in 2009: A Review and Future Visions. Journal of Educational Data Mining, 1 (1), 3-17, 2009.

BARANAUSKAS, J.A., Monard, M.C. reviewing some learning machine concepts and methods. Relatórios técnicos do ICMC, n.102, p.52 USP, São Carlos, 2000.

BEZERRA C, Scholz r., Adeodato P, Lucas T., and Ataide I., “Evasão Escolar: Aplicando Mineração de Dados para Identificar Variáveis Relevantes,” Anais do XXVII Simpósio Bras. Informática na Educ. (SBIE 2016), vol. 1, no. CBIE, p. 1096, 2016

BAYER, J. et al. “Predicting drop-out from social behaviour of students.” In: Proceedings of the 5th International Conference on Educational Data Mining - EDM 2012, p. 103-109. ISBN 978-1-74210-276-4, 2012.

BENNEDSEN, J., Caspersen, M. E.. Failure rates in introductory programming. SIGCSE Bull., 39, 32–36, URL: http://doi.acm.org/10.1145/1272848.1272879. doi:10.1145/1272848.1272879, 2007.

BREIMAN, L.; Friedman, J. H.; Olshen, R. A.; Stone, C.J. Classification and regression trees. Belmont, CA: Wadsworth International, 1984. 358 p

BUCKINGHAM, SHUM, ; Ferguson, R. Social Learning Analytics. Educational Technology & Society, 15 (3), 3–26. 2012.

BOUCKAERT, R. R.; FRANK, E.; HALL, M.; et al. WEKA Manual for Version 3-6-5. .

BURGOS, C; Campanario M. L., D. de la Peña, J. A. Lara, D. Lizcano, and M. A. Martínez, “Data mining for modeling students’ performance: A tutoring action plan to prevent academic dropout,” Comput. Electr. Eng., vol. 66, pp. 541–556, 2018.

BYDZOVSKA, H. A comparative analysis of techniques for predicting student performance (pp. 306e311). International Educational Data Mining Society.2016

CAMBRUZZI, W., Rigo, S., and Barbosa, J. Dropout prediction and reduction in distance education courses with the learning analytics multitrail approach. Journal of Universal Computer Science, 21(1):23–47.2015.

CARBONELL, J.R. “AI in CAI: an artificial intelligence approach to computer- assisted instruction”, IEEE Transactions on Man-Machine Systems, 11(4): 190-202, 1970.

CARUANA, R.; Niculescu-Mizil, A. An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd International Conference on Machine Learning.New York, NY, USA: ACM, 2006. (ICML '06), p. 161_168. ISBN 1- 59593- 383-2. Disponível em: http://doi.acm.org/10.1145/1143844.1143865, 2006.

CENDROWSKA, J. PRISM: An algorithm for inducing modular rules. J. Cendrowska, “Prism: An algorithm for inducing modular rules,” International Journal of Man-Machine Studies, Vol. 27, No. 4, pp. 349–370, 1987.

CHATTI, M.A., Dyckhoff, A.L., Schroeder, U. and Thus, H. Learning analytics: a review of the state of the art and future challenges’, International Journal of Technology Enhanced Learning. 2012.

CHAPMAN, P. et al, 2000. CRISP-DM 1.0 - Step-by-step data mining guide. Accessed from http://www.crisp-dm.org/CRISPWP-0800.pdf

CHAWLA, N. V., Bowyer, K. W., Hall, L. O., e Kegelmeyer, W. P. SMOTE: Synthetic minority over-sampling technique. J. Artif. Intell. Res. (JAIR), 16, 321–357. 2002.

CHRYSAFIADI, K.; Virvou, M. “Student modeling approaches: A literature review for the last decade.” Expert Systems with Applications, 40(11): 4715-

4729 DOI: 10.1016/j.eswa.2013.02.007. 2013.

CHUI, K.T.,Fung, D. C. L., Lytras, M.D., Lam, T. M. Predicting at-risk university students in a virtual learning environment via a machine learning algorithm. Computers in Human Behavior, vol. 73, pp. 247–256, 2018.

CONIJN, R., Snijders C., Kleingeld A ., Matzat U. Predicting Student Performance from LMS Data: A Comparison of 17 Blended Courses Using Moodle LMS. IEEE Transactions on Learning Technologies.. 2017.

CORTES, C and Vapnik, V. “Support-Vector Networks,” Mach. Learn., 1995.

COSTA, E. et al. Mineração de dados educacionais: Conceitos, técnicas, ferramentas e aplicações. Jornada de Atualização em Informática na Educação - JAIE, v. 02, n. 02,p. 03, 2012. ISSN 23167734. Disponível em: http://www.brie.org/pub/index.php/pie/article/view/2341/2096 .

COSTA, E. B., Fonseca, B., Santana, M. A., Araújo,F. F. de and Rego J., “Evaluating the effectiveness of educational data mining techniques for early prediction of students’ academic failure in introductory programming courses,” Computers in Human Behavior, vol. 73, pp. 247–256, 2017.

COVER T . M; Hart P .E.; Nearest neighbor pattern classification. IEEE Transactions on Information Theory 13 (1): 21–27. doi:10.1109/TIT.1967.1053964.1967.

De ARAÚJO, F. F. “Uma Abordagem Computacional para Assistência Adaptativa a Estudantes e Professores em Ambientes de Aprendizagem Online.” Proposta de tese apresentada à Coordenação do Curso de Pós-Graduação em Ciência da Computação da UFCG, 2016.

D. THAMMASIRI, D. Delen, P. Meesad, and N. Kasap, “A critical assessment of imbalanced class distribution problem: The case of predicting freshmen student attrition,” Expert Systems with Applications, vol. 41, pp. 321–330, 2014.

ER, E. “Identifying at-risk students using machine learning techniques: A case study with is 100.” In: International Journal of Machine Learning and Computing. Singapore: IACSIT Press, 2012. p. 476-481. ISBN 978-1-4503-

2469-4. Disponível em: http://doi.acm.org/10.1145/2554850.2555135.

ESICHAIKUL, V.; Lamnoi, S.; Bechter, C. “Student Modelling in Adaptive E- Learning Systems.” Knowledge Management & E-Learning: An International Journal, Vol. 3, No. 3, 2011.

ESSALMI, F.; Ben Ayed, L. J.; Jemni, M., and Graf, S., “Generalized metrics for the analysis of E-learning personalization strategies,” Computers in Human Behavior, vol. 48, pp. 310–322, Jul. 2015.

FACELI, K., Lorena, A. C., Gama, J., e Carvalho, A. C. P. L. F. Inteligência artificial: Uma abordagem de aprendizado de máquina. Rio de Janeiro: LTC, 2:192. 2011

FAYYAD, U. M., Piatetsky Shapiro, G., Smyth, P. & Uthurusamy, R. “Advances in Knowledge Discovery and Data Mining”, AAAIPress, The Mit Press. 1996.

FRANK, E., Hall, M.A., Witten, I. H. “The WEKA Workbench Data Mining:Practical Machine Learning Tools and Techniques,” Morgan Kaufmann, Fourth Ed., p. 128, 2016.

FREUND, Y; Mason, K. The Alternating Decision Tree Learning Algorithm. In Proc. 16th International Conf. on Machine Learning, pp. 124-133 Key: citeulike:8970123. 1999

GAŠEVIĆ, D., Dawson, S., Siemens G. “Let’s not forget: Learning analytics are about learning.” TechTrends, January 2015, Volume 59, Issue 1, pp 64-71, 2015.

GAŠEVIĆ D., Dawson S., Rogers T., and Gasevic D. Learning analytics should not promote one size fits all: The effects of instructional conditions in predicting learning success. The Internet and Higher Education. 28, (2016), 68–84. 2016.

GOEBEL, M., Gruenwald, L. A survey of data mining and knowledge Discovery software tools. ACM SIGKDD, San Diego, v.1, n.1. p. 20-33, 1999.

GOTARDO, R., Cereda, P. R. M., and Junior, E. R. H. Predição do desempenho do aluno usando sistemas de recomendação e acoplamento de classificadores. XXIV Simpósio Brasileiro de Informática na Educação (SBIE

2013).

GOTTARDO, E., Kaester, C., and Noronha, R. V. Previsão de desempenho de estudantes em cursos ead utilizando mineração de dados: uma estratégia baseada em séries temporais. Anais do XXIII SBIE. 2012.

GOTTARDO, E.; Noronha, R. V.; and Kaester, C. “Estimativa de Desempenho Acadêmico de Estudantes: Análise da Aplicação de Técnicas de Mineração de Dados em Cursos a Distância.” Revista Brasileira de Informática na Educação, 2014. - 1 : Vol. 22. 2014.

GURULER H., Istanbullu A., Karahasan M., A new student performance analysing system using knowledge discovery in higher educational databases--Computers & Education, Elsevier, 2010.

HAMALAINEN, W. and Vinni, M. Classifiers for educational data mining. Chapman & Hall/CRC Data Mining and Knowledge Discovery Series, 2011.

HAN, J; Kamber, M; PEI, J.Data Mining: concepts and techniques. 3rd Ed. ISBN: 978-0-12-381479-1.Ed. Elsevier.2011.

HANKS, B., McDowell, C., Draper, D., Krnjajic, M. Program quality with pair programming in cs1. SIGCSE Bull.,36,176–180. URL: http:// doi.acm.org/10.1145/1026487.1008043. doi:10.1145/1026487.1008043. 2004.

HAYKIN, S. Redes Neurais: Princípios e prática. 2.ed. Porto Alegre: Bookman, 2001. 900p.

HSSINA B., Merbouha A., Ezzikouri H. and Erritali M., “A comparative study of decision tree ID3 and C4.5” International Journal of Advanced Computer Science and Applications(IJACSA), Special Issue on Advances in Vehicular Ad Hoc Networking and Applications. http://dx.doi.org/10.14569/SpecialIssue.2014.040203. 2014.

HOLZHÜTER M., Frosch-Wilke D., Klein U., Exploiting Learner Models Using Data Mining for E-Learning: A Rule Based Approach. In: Peña-Ayala A. (eds) Intelligent and Adaptive Educational-Learning Systems. Smart Innovation, Systems and Technologies, vol 17. Springer, Berlin, Heidelberg. 2013.

HU, Y. -H., Lo, C.-L., and Shih, S.-P. (2014). Developing early warning systems to predict students’ online learning performance. Computers in Human

Behavior, 36:469 – 478.

HUNG, J. L., Wang, M., Wang, S., Abdelrasoul, M., y. li, and He, W. (2016). Identifying at-risk students for early interventions? a time-series clustering approach. IEEE Transactions on Emerging Topics in Computing, 99:45 – 55.

HUXLEY. THE HUXLEY . 2018. http://www.thehuxley.com/. Acessado em Março- 2018.

IEPSEN, E., Bercht, M., & Reategui, E. Detection and assistance to students who show frustration in learning of algorithms. Em: Frontiers in Education Conference, 2013. IEEE (pp. 1183–1189). doi:10.1109/FIE.2013. 6685017. 2013.

JAIN, D.; Kedia, A.; Singla, R.; and Sonawane, S. (2015). “Recommendation Techniques for Adaptive E-learning.” Advances in Computer Science and Information Technology (ACSIT)Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 1; January-March, 2015 pp. 7-12.

J. W. You, “Identifying significant indicators using LMS data to predict course achievement in online learning,” Internet and Higher Education, vol. 29, pp. 23–30, 2016.

JOHN, G. H. e Langley, P. (1995). Estimating continuous distributions in bayesian classifiers. In Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, pages 338–345. Morgan Kaufmann Publishers Inc,1995.

KAMPFF, A. J. C. “Mineração de Dados Educacionais para Geração de Alertas em Ambientes Virtuais de Aprendizagem como Apoio à Prática Docente.” Revista Informática na Educação: teoria & prática. e-ISSN: 1982-1654, v. 12, n. 2 .2009.

KHALIL, Hanan and Martin Ebner. 2014. MOOCs completion rates and possible methods to improve retention –a literature review. In EdMedia:World Conference on Educational Multimedia, Hypermedia and Telecommunications, AACE. 1236-1244.

KEARSLEY, G. (eds). Artificial Intelligence and Instruction: applications and methods, Addison-Wesley Publishing Company, 1987.

KEARSLEY, G. “Intelligent Agents and Instructional Systems: Implications of a New Paradigm.” Journal of Artificial Intelligence in Education, 4(4) 295-304, 1993.

KHOBRAGADE, L., Magadik, P., Students academic failure prediction using data mining. International Journal of Advanced Research in Computer and Communication Engineering Vol. 4, Issue 11, Novembro, 2015.

KOEDINGER, K. R.; Brunskill, E.; Baker, R. J.d.; McLaughlin. E. A.; Stamper, J. “New Potentials for Data-Driven Intelligent Tutoring System Development and Optimization.”AI Magazine, Vol 34, No 3, 2013. DOI: http://dx.doi.org/10.1609/aimag.v34i3.2484 .

KOLLER, Daphne, Andrew Ng, Chuong Do, and Zhenghao Chen, Retention and intention in massive open online courses: In depth. Educause Review, 48(3). 62–63, June 2013. Disponível em : http://er.educause.edu/articles/2013/6/retention-and-intention-in-massive-open-online-courses

KOTSIANTIS, S. B. (2009). Educational data mining: a case study for predicting dropout- prone students. IJKESDP, 1(2): 101–111.

LIÑÁN, L.C., Perez, A.A. Educational data mining and learning analytics: differences, similarities, and time evolution. RUSC. Universities and Knowledge Society Journal, 12(3):98–112, 2015.

LYKOURENTZOU, I., Giannoukos I., Nikolopoulos V.Dropout prediction in e-learning courses through the combination of machine learning techniques. Computers & Education. Volume 53, Issue 3, , P. 950-965 , 2009.

MACFADYEN, L. and Dawson, S. Mining LMS data to develop an “early warning system” for educators: A proof of concept. Computers & Education, 54(2): 588–599, 2010.

MANHÃES, L. M. B.; da Cruz, S. M. S.; Zimbrão, G “Wave. An architecture for predicting dropout in undergraduate courses using EDM.” In Proceedings of the 29th Annual ACM Symposium on Applied Computing SAC ’14 (pp. 243–247). New York, NY , USA: ACM. . (2014). URL: http://doi.acm.org/ 10.1145/2554850.2555135. doi:10.1145/2554850.2555135.

MARBOUTI, F., Diefes-Dux, H. A., Madhavan, K. Models for early prediction of at-risk students in a course using standards-based grading. Computers & Education, vol. 103, 1-15, 2016.

MARQUEZ-VERA, C. , Morales, C.; Soto, S. Predicting school failure and dropout by using data mining techniques. Tecnologias del Aprendizaje, IEEE Revista Iberoamericana de, v. 8, n. 1, p. 7-14, Feb 2013. ISSN 1932-8540.

MARQUEZ-VERA, C., Cano, A., Romero, C., Ventura, S. Predicting student failure at school using genetic programming and different data mining approaches with high dimensional and imbalanced data. Applied Intelligence. 38. 315-330. doi:10.1007/s10489-012-0374-8. 2013.

MARQUEZ-VERA, C., Cano A. , Romero, C., A. Y. M. Noaman, H. Mousa Fardoun, and S. Ventura, “Early dropout prediction using data mining: a case study with high school students,” Expert Systems, vol. 33, no. 1, pp. 107–124, 2 2016.

MARTINHO, V.; Nunes, C.; Minussi, C.R. Prediction of school Dropout Risk Group Using Neural Network. Proceedings of the 2013 Federated Conference on Computer Science and Information Systems, 111-114, 2013.

MARTINHO, V., Nunes, C., Minussi, C.R. An Intelligent System for Prediction of School Dropout Risk Group in Higher Education Classroom Based on Artificial Neural Networks. IEEE 25th International Conference on Tool with Artificial Inteligence.2013.doi: 10.1109/ICTAI.2013.33

MARTINS, A. C., Faria, L.; Vaz de Carvalho, C., ; Carrapatoso, E. (2008). User Modeling in Adaptive Hypermedia Educational Systems. Educational Technology ; Society, 11 (1), 194-207.

MARTINA A. Rau, Vincent Aleven, and Nikol Rummel. Intelligent tutoring systems with multiple representations and self-explanation prompts support learning of fractions. In Proceedings of the Conference on Artificial Intelligence in Education, 2009.

MCGETTRICK, A.; Boyle, R.; Ibbett, R.; Lloyd, J., Lovegrove, G.; and Mander, K. Grand Challenges in Computing: Education – A Summary. The Computer Journal vol. 48. N. 1. The British Computer Society, 2005. DOI:

10.1093/comjnl/bxh064.

MITCHELL, T.M. Machine Learning. 1. ed. New York. Mcgraw-Hill, Inc., 1997.

MORALES-RODRÍGUEZ, M. L.; Ramírez-Saldivar, J. A.; Sánchez-Solís, J.P.; Hernández- Ramírez, A.; Martínez-Flores, J.A. (2012). “Design of an Intelligent Agent for Personalization of Moodle’s Contents.” 11th Mexican International Conference on Artificial Intelligence, WILE 2012: Fifth Workshop on Intelligent Learning Environments.

O’DONNELL, E.; Lawless, S.; Sharp, M. and Wade, V. P. “A Review of Personalised E- Learning: ,” Int. J. Distance Educ. Technol., vol. 13, no. 1, pp. 22–47, 2015.

PAES, R. de B.; Malaquias, R.; Guimarães, M.; e Almeida; H. “Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores”. II Congresso Brasileiro de Informática na Educação (CBIE 2013), Campinas - SP, 2013.

PAPAMITSIOU, Z., & Economides, A. (2014). Learning Analytics and Educational Data Mining in Practice: A Systematic Literature Review of Empirical Evidence. Educational Technology & Society, 17 (4), 49–64.

PEÑA-AYALA, A. “Educational data mining: A survey and a data mining-based analysis of recent works.” Expert systems with applications, 41(4):1432–1462, 2014.

PENSTEIN, Carolyn Rosé, Ryan Carlson, Diyi Yang, Miaomiao Wen, Lauren Resnick, Pam Goldman, and Jennifer Sherer. 2014. Social Factors that Contribute to Attrition in MOOCs. In Proc. of the 1st ACM Conference on Learning at Scale (L@S), Atlanta, 2014. http://dx.doi.org/10.1145/2556325.2567879.

QUINLAN, J. R. Induction of decision trees. Mach. Learn., Kluwer Academic Publishers,Hingham, MA, USA, v. 1, n. 1, p. 81_106. ISSN 0885-6125.1986.

QUINLAN J. R., C4.5: Programs for Machine Learning. 1993.

RAADT, M. d., “A Review of Australasian Investigations into Problem

Solving and the Novice Programmer," Computer Science Education, vol. 17, pp. 201-213, 2007.

REDA M. A., Nahla F. O. and Abdelmgeid A .A. Predicting and Analysis of Students' Academic Performance using Data Mining Techniques. International Journal of Computer Applications 182(32):1-6, December 2018.

ROCHA, R. H. S.; Costa, E de B.; Brito, P. H. da S. "Improving Construction and Maintenance of Agent-based Applications through an Integration of Shell and Software Framework Approaches." Encontro Nacional de Inteligência Artificial ,2012.

ROMERO, C.; Ventura, S. Educational data mining: A review of the state of the art. IEEE Transactions on Systems, Man, and Cybernetics, Part C, 40(6):601–618, 2010.

ROY, S; Garg, A , "Predicting academic performance of student using classification techniques," in 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics, UPCON 2017, vol. 2018–Janua, pp. 568–572,2018.

RUSSELL, S. and Norvig, P. Artificial Intelligence: A Modern Approach. Prentice Hall, 2009.

SANDOVAL, A.; Gonzalez, C.; Alarcon, R.; Pichara, K; Montenegro, M. Centralized student performance prediction in large courses based on low- cost variables in an institutional context. The Internet and Higher Education 37 (2018) 76–89.

SELF. J. A. "Student Models in Computer-aided Instruction", International Journal of Man-Machine Studies, 6:261- 276, 1974.

SHAHIRI, A. M., Husain, W., Rashid, N. A. A review on predicting student's performance using data mining techniques. Procedia Computer Science, 72, 414– 422. 2015

SIBANDA, L., Iwu, C, and Benedict, H. Factors Influencing Academic Performance of University Students. Demography and social economy. 103-115. 10.15407/dse2015.02.103. 2015.

SILVA, D. H.; Dorça, F. A. (2014). Uma Abordagem Automática para

Personalização do Processo de Ensino Baseada em Estilos de Aprendizagem em Sistemas Adaptativos e Inteligentes para Educação a Distância. Revista Brasileira de Informatica na Educação (RBIE), v. 22, n. 2, p. 1–15.

SINGH S.,Gupta P. , “Comparative Study Id3 , Cart And C4 . 5 Decision Tree Algorithm : A Survey,” Int. J. Adv. Inf. Sci. Technol., 2014.

SLEEMAN, D. ; Brown, J,.S.(eds) Intelligent Tutoring systems Computers and people series. Academic Press - Corrnell University, Londres (RU), 1982.

SPOELSTRA, H.; Rosmalen, P. V.; Houtmans, T. and Sloep P. “Team Formation Instruments to Enhance Learner Interactions in Open Learning Environments”. Computers in Human Behavior, vol. 45, pp.11–20, 2015.

TAN, P.-H., Ting, C.-Y., Ling, S.-W. (2009). Learning dificulties in programming courses: Undergraduates’ perspective and perception. In Computer Technology and Development, 2009. ICCTD ’09. International Conference on (pp. 42–46). volume 1.doi:10.1109/ICCTD,2009.

VAPNIK, V. N. The Nature of Statistical Learning Theory. New York, NY, USA:Springer-Verlag New York, Inc. ISBN 0-387-94559-8., 1995

WATSON, C., & Li, F. W. Failure rates in introductory programming revisited. Proceedings of the 2014 Conference on Innovation and Technology in Computer Science Education ITiCSE ’14 (pp. 39–44). New York, NY, USA: ACM. URL: http://doi.acm.org/10.1145/2591708.2591749. doi:10.1145/2591708.2591749. 2014.

WENGER, E. Artificial Intelligence and Tutoring Systems: Computacional and cognitive approaches to the Communication of Knowledge. Los Altos, CA: Morgan Kaufmann,1987.

WITTEN, I. H.; Frank, E.; Hall, M. A. “Data Mining: Practical Machine Learning Tools and Techniques.” 3rd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers. Inc. ISBN 0123748569, 9780123748560, 2011.

WITTEN, I.H., and Frank E., Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, San Francisco, 2005.

WOOLF, B.P. “Building Intelligent Interactive Tutors: Student-centered

strategies for revolutionizing e-learning. Morgan Kaufmann, San Francisco, 2008.

WOOLF, B.P.; Lane, H. C.; Chaudhri, V. K., and Kolodner, J. L. “AI Grand Challenges for Education,” AI Magazine, vol. 34, no. 4. pp. 66–84, 2013.

WU, X. et al. Top 10 algorithms in data mining. Knowledge and Information Systems, Springer-Verlag, v. 14, n. 1, p. 1_37, 2008. ISSN 0219-1377. Disponível em:http://dx.doi.org/10.1007/s10115-007-0114-2, 2008.

YACEF, K. The logic-ita in the classroom: A medium scale experiment. International Journal of Artificial Intelligence in Education, 2005.

YASSEIN, N.A., Helali R. G. M, and Mohomad S. B., “Predicting Student Academic Performance in KSA using Data Mining Techniques,” J. Inf. Technol. Softw. Eng., vol. 07, no. 05, 2017.

ZAFFAR, Maryam, Hashmani Manzoor Ahmed, K.S. Savita, Syed Sajjad Hussain Rizvi. A Study of Feature Selection Algorithms for Predicting Students Academic Performance. (IJACSA) International Journal of Advanced Computer Science and Applications, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 5, 2018.

---

i